

VIŠJA STROKOVNA ŠOLA ACADEMIA

MARIBOR

**RAZVOJ SODOBNIH INFORMACIJSKIH
REŠITEV V MALOKODNEM RAZVOJNEM
OKOLJU ORACLE APEX NA PRIMERU
RAZŠIRITVE FUNKCIONALNOSTI OBSTOJEČE
APLIKACIJE**

Kandidatka: Nuša Hertiš

Vrsta študija: študentka izrednega študija

Študijski program: Programsko inženirstvo

Mentor predavatelj: mag. Ervin Schaff

Mentor v podjetju: Tine Zorko, dipl. inž. inf. in tehnol. kom.

Lektor: Kaja Podgoršek, mag. prof. slov. jez. in knjiž. in mag. prof. ang.

Maribor, 2025

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Podpisana Nuša Hertiš sem avtorica diplomskega dela z naslovom Razvoj sodobnih informacijskih rešitev v malokodnem razvojnem okolju Oracle Apex na primeru razširitve funkcionalnosti obstoječe aplikacije, ki sem ga napisala pod mentorstvom mag. Ervina Schaffa.

S svojim podpisom zagotavljam, da:

- je predloženo delo izključno rezultat mojega dela,
- sem poskrbel/a, da so dela in mnenja drugih avtorjev, ki jih uporabljam v predloženi nalogi, navedena oz. citirana skladno s pravili Višje strokovne šole Academia Maribor,
- se zavedam, da je plagiatorstvo – predstavljanje tujih del oz. misli, kot moje lastne kaznivo po Zakonu o avtorski in sorodnih pravicah (Uradni list RS, št. 16/07 – uradno prečiščeno besedilo, 68/08, 110/13, 56/15 in 63/16 – ZKUASP); prekršek pa podleže tudi ukrepom Višje strokovne šole Academia Maribor skladno z njenimi pravili,
- skladno z 32.a členom ZASP dovoljujem Višji strokovni šoli Academia Maribor objavo diplomskega dela na spletnem portalu šole.

Maribor, junij 2025

Podpis študenta:

ZAHVALA

Zahvaljujem se podjetju Zavarovalnica Sava, d. d., v katerem sem opravljala praktično izobraževanje in napisala diplomsko delo.

Iskreno se zahvaljujem svojemu mentorju v podjetju dipl. inž. inf. in tehnol. kom. Tinetu Zorku za strokovno pomoč in nasvete pri izdelavi diplomskega dela. Prav tako se zahvaljujem univ. dipl. oec. Matjažu Cvaru ob tehnični pomoči.

Iskrene zahvale izrekam mentorju predavatelju mag. Ervinu Schaffu za strokovne nasvete in ažurno pomoč.

Zahvale izrekam svoji lektorici mag. prof. slov. jez. in knjiž. in mag. prof. ang. Kaji Podgoršek, ki ni le lektorirala diplomskega dela, ampak je o njem tudi strpno poslušala in nudila podporo med pisanjem.

Na koncu bi se rada zahvalila še svoji družini, ki je vedno postavljala moje želje po izobrazbi na prvo mesto in mi nudila podporo, ko sem prišla do ovir ter se z mano veselila, ko sem jih premagala.

POVZETEK

V diplomskem delu smo se osredotočili na razvoj informacijskih rešitev v malokodnem razvojnem okolju Oracle Apex, kjer smo naredili nadgradnjo obstoječe centralne rešitve za upravljanje identitet in dostopov IAM.

V teoretičnem izhodišču so predstavljena malokodna razvojna okolja. Izpostavili smo okolje Oracle Apex, ki je osnovano na Oracle relacijski podatkovni bazi in uporablja SQL in PL/SQL programska jezika. Predstavili smo osnovne lastnosti, zgradbo, delovanje in izgled okolja. Opisali smo vlogo umetne inteligence na področju razvoja informacijskih rešitev v malokodnih okoljih.

Analizirali smo glavne lastnosti sistemov za upravljanje z identitetami in dostopi, pri čemer smo izpostavili ključne poslovne procese, ki jih ti sistemi podpirajo. Prav tako smo opisali revizijske sledi in predstavili njihov pomen.

Analizirali smo obstoječo IAM centralno rešitev v malokodnem razvojnem okolju Oracle Apex. Pri tem smo se osredotočili na zgradbo in osnovne poslovne procese v IAM centralni rešitvi. V izbran poslovni proces smo vgradili beleženje revizijskih sledi z razširitvijo obstoječe programske knjižnice. Prav tako smo v malokodno razvojno okolje vgradili dve generativni AI storitvi, in sicer Cohere in OpenAI, ter dodali AI pomočnika, ki na osnovi podanega poljudnega pozivnega sporočila generira poizvedbe in odgovore o revizijskih sledeh. Odgovore izbranih storitev smo nato med seboj primerjali.

Med nadgradnjo informacijske rešitve smo ugotovili, da malokodna razvojna okolja omogočajo hiter razvoj preprostejših rešitev ali prototipov aplikacij brez potrebnega znanja o programiranju. Za zahtevnejše nadgradnje pa je potrebno znanje programskega jezika. V našem primeru predvsem znanje o SQL in PL/SQL jeziku.

Potrdili smo, da IAM centralna rešitev omogoča transparenten in kontroliran dostop do funkcionalnosti v informacijskih rešitvah. Prav tako smo potrdili, da zgodovinske tabele same po sebi ne omogočajo pregleda nad izvajanjem poslovnih dogodkov.

Po implementaciji generativnih AI rešitev smo s pomočjo kriterijev učinkovitosti (točnost/kvaliteta, kapaciteta/kompleksnost in izraba virov) analizirali in ugotovili, da je storitev OpenAI učinkovitejša od storitve Cohere na primeru iz tega diplomskega dela.

Ključne besede: *Apex, revizijska sled, podatkovne baze, malokodno okolje, umetna inteligenca.*

ABSTRACT

DEVELOPMENT OF MODERN INFORMATION SOLUTIONS IN THE LOW-CODE PLATFORM ORACLE APEX ON THE EXAMPLE OF ENHANCING THE FUNCTIONALITY OF AN EXISTING APPLICATION

This thesis explores the development of modern information solutions in the low-code development platform Oracle Apex, with a focus on enhancing an existing centralized identity and access management (IAM) solution.

In the theoretical part, we introduce the concept of low-code platforms, highlighting Oracle Apex, which is based on the Oracle relational database and uses the SQL and PL/SQL programming languages. We describe its core features, architecture, functioning and user interface. We also examined the role of artificial intelligence (AI) in the development of information systems in low-code environments.

We analyzed key aspects of IAM systems, highlighting key business processes supported by this system. In addition, we discussed the importance of audit trails for ensuring traceability and accountability. We analyzed the existing central IAM solution within the Oracle Apex environment, focusing on its structure and core business processes. We integrated audit trails into a selected business process by extending the existing software library. Furthermore, we implemented two generative AI services, Cohere and OpenAI, into the low-code environment and added an AI assistant that generates queries and answers in natural language regarding audit trails based on a system prompt. The responses of the selected services were then compared.

During the upgrade of the information solution, we found that low-code development environments enable rapid development of simpler solutions or application prototypes without requiring programming knowledge. However, more complex upgrades require knowledge of programming languages. Specifically, in our case, knowledge of SQL and PL/SQL languages.

We confirmed that the central IAM solution enables transparent and controlled access to functionalities in information systems. We also confirmed that historical tables alone do not provide sufficient insight into the business processes. After implementing generative AI solutions, we evaluated them using performance criteria (accuracy/quality, capacity/complexity and resource usage). We found that the OpenAI service outperformed Cohere service in the context of this thesis.

Keywords: *Apex, audit trail, database, low-code platform, artificial intelligence.*

KAZALO VSEBINE

1	UVOD.....	13
1.1	Opis področja in opredelitev problema	13
1.2	Namen, cilji in osnovne trditve.....	13
1.3	Predpostavke in omejitve	14
1.4	Uporabljene raziskovalne metode	15
2	MALOKODNA RAZVOJNA OKOLJA	16
2.1	Značilnosti in namen	16
2.2	Prednosti	17
2.3	Razvoj in sodobni trendi.....	19
2.3.1	Podpora umetne inteligence	19
2.3.2	AI pogovorni sistemi	20
3	ORACLE APEX.....	21
3.1	Lastnosti	21
3.2	Zgradba in delovanje	22
3.2.1	Oracle podatkovna baza in PL/SQL	23
3.2.2	Aplikacijski strežnik ORDS	24
3.2.3	Večnivojska arhitektura	25
3.3	Možnosti namestitve v različnih okoljih	25
3.4	Upravljanje okolja	26
3.5	Upravljanje aplikacij.....	28
3.5.1	Administracija	30
3.6	Rešitve z uporabo umetne inteligence	31
4	IAM	32
4.1	Opis IAM rešitev	32
4.2	Osnovne funkcionalne zahteve.....	33
4.3	Avtentikacija in avtorizacija.....	34
4.3.1	Avtorizacijske sheme.....	34
5	REVIZIJSKE SLEDI.....	36
5.1	Zahteve	36

5.2	Struktura zapisa	37
5.3	Vpliv beleženja na delovanje aplikacij	38
5.4	Skrbništvo nad podatki	38
6	OBSTOJEČA IAM REŠITEV V OKOLJU ORACLE APEX	39
6.1	Obseg in omejitve	39
6.2	Podatkovni model	40
6.3	Zgradba aplikacije	41
6.3.1	Uporaba IAM centralne rešitve	41
6.3.2	Avtorizacijska shema obstoječe IAM rešitve	42
6.4	Uporabniški profili	43
6.5	Osnovni poslovni procesi	45
7	NADGRADNJA IAM APEX APLIKACIJE	46
7.1	Identifikacija dogodkov za zapis revizijske sledi	46
7.2	Zapisovanje	47
7.2.1	Tabela dnevniških zapisov in revizijske sledi	48
7.2.2	Programiranje v malokodnem razvojnem okolju Oracle Apex	52
7.3	Hranjenje	52
7.4	Dostop in upravljanje	53
7.4.1	Avtorizacijska shema v IAM centralni rešitvi	53
7.5	Pregledi	54
7.6	AI pomočnik	55
7.6.1	Implementacija AI pomočnika	55
7.6.2	Kriteriji učinkovitosti	57
7.6.2.1	Točnost/kvaliteta	58
7.6.2.2	Kapaciteta/kompleksnost	61
7.6.2.3	Izraba virov	63
7.6.3	Interpretacija	65
8	SKLEP	66
9	VIRI IN LITERATURA	69

KAZALO TABEL

Tabela 1: Legenda ocen in odgovorov	58
Tabela 2: Kriterij točnost/kvaliteta	61
Tabela 3: Kriterij kapaciteta/kompleksnost	63
Tabela 4: Vrednotenje AI storitev	65

KAZALO SLIK

Slika 1: Malokodne platforme	17
Slika 2: Mnenja podjetij Fortune 500 firms.....	18
Slika 3: Diagram arhitekture Oracle Apex	22
Slika 4: Oracle podatkovna baza	23
Slika 5: ORDS	24
Slika 6: Večnivojska arhitektura	25
Slika 7: Oracle Apex domača stran	26
Slika 8: Oracle Apex graditelj aplikacij	26
Slika 9: Oracle Apex delavnica SQL.....	27
Slika 10: Oracle Apex skupinsko razvoj	27
Slika 11: Oracle Apex galerija aplikacij.....	28
Slika 12: Oracle Apex graditelj aplikacij	28
Slika 13: Oracle Apex oblikovalnik strani	29
Slika 14: Oracle Apex skupne komponente	30
Slika 15: Oracle Apex administracija.....	30
Slika 16: RAG v Oracle Apex	31
Slika 17: IAM funkcionalnosti	33
Slika 18: Čarovnik za avtorizacijsko shemo	35
Slika 19: Oracle Apex tipi avtorizacijskih shem	35
Slika 20: GDPR na kratko	37
Slika 21: Omejen podatkovni model IAM centralne rešitve	40
Slika 22: Domača stran IAM rešitve	41
Slika 23: Avtorizacijska shema - PL/SQL procedura.....	42
Slika 24: PL/SQL telo funkcije avtorizacijske sheme IAM rešitve.....	43
Slika 25: Urejanje uporabnika	44
Slika 26: Dodeljeni profili in privilegiji uporabnika	44
Slika 27: Podatkovni model: Dodelitev profila uporabniku.....	47
Slika 28: Zgodovinska tabela	48
Slika 29: Čarovnik za ustvarjanje tabele	49
Slika 30: DDL prožilec.....	49
Slika 31: DDL paket LOG_PCK.....	50
Slika 32: DDL paket REV_PCK	50

Slika 33: Koda procedure	52
Slika 34: Urejevalnik strani - avtorizacijska shema	53
Slika 35: Dostop ni odobren	54
Slika 36: Čarovnik ustvarjanje nove strani	54
Slika 37: Prikaz revizijskih sledi	55
Slika 38: Faza 1: Generativna AI storitev	56
Slika 39: Faza 2: AI konfiguracija	56
Slika 40: Faza 3: RAG za AI konfiguracijo	57
Slika 41: Faza 4: AI pomočnik s pripadajočimi dinamičnimi akcijami in gumbi	57
Slika 42: AI pomočnik - 1. vprašanje	59
Slika 43: AI pomočnik - 2. vprašanje	59
Slika 44: AI pomočnik - 3. vprašanje	59
Slika 45: AI pomočnik - 4. vprašanje	60
Slika 46: AI pomočnik - 5. vprašanje	60
Slika 47: AI pomočnik - 6. vprašanje	60
Slika 48: AI pomočnik - 7. vprašanje	62
Slika 49: AI pomočnik - 8. vprašanje	62
Slika 50: AI pomočnik - 9. vprašanje	62
Slika 51: Cohere - poraba tokenov	64
Slika 52: OpenAI - poraba tokenov	64

OKRAJŠAVE IN KRATICE

4GL – Fourth generation languages (četrta generacija programskih jezikov)

AAA – Authentication, authorization and accounting framework (okvir za avtentikacijo, avtorizacijo in evidentiranje)

AI – Artificial intelligence (umetna inteligenca)

AJAX – Asynchronous JavaScript and XML (asinhroni JavaScript in XML)

API – Application programming interace (aplikacijski programski vmesnik)

BPE – Byte-pair encoding (kodiranje z združevanjem bajtov)

DBMS – Database management system (sistem za upravljanje podatkovnih baz)

DDL – Data definition language (jezik za definicijo podatkov)

GDPR – General data protection regulation (splošna uredba za varstvo podatkov)

HTTPS – HyperText transfer protocol secure (varen protokol za izmenjavo hiperteksta)

IAM – Identity and access management (upravljanje identitet in dostopov)

IGA – Identity Governance and administration (upravljanje in nadzor identitet)

ISO – International Organization for Standardization (mednarodna organizacija za standardizacijo)

IEC – International Electrotechnical Commission (mednarodna elektrotehnična komisija)

ITDR – Identity threat detection and response (odkrivanje in odziv na grožnje identitet)

Java EE – Java platform, enterprise edition (Java strežniška različica)

JSON – JavaScript Object Notation (JavaScript objektna notacija)

MFA – Multifactor Authentication (multifaktorna avtentikacija)

NLP – Natural language processing (procesiranje naravnega jezika)

NLU – Natural language understanding (razumevanje naravnega jezika)

NLG – Natural language generation (generiranje naravnega jezika)

OCI – Oracle Cloud Infrastructure (Oracle oblačna infrastruktura)

ORDS – Oracle REST data services (Oracle REST podatkovne storitve)

PaaS – Platform as a Service (platforma kot storitev)

PL – Procedural language (proceduralni jezik)

PWA – Progressive web apps (progresivne spletne aplikacije)

RAG – Retrieval-augmented generation (generiranje z uporabo pridobivanja informacij)

RBA – Risk-based authentication (avtentikacija na osnovi tveganja)

RBAC – Role based access control (nadzor dostopa na podlagi vlog)

SaaS – Software as a Service (programska oprema kot storitev)

SSO – Single sign on (enotna prijava)

SUPB – Sistem za upravljanje podatkovnih baz

SQL – Structured query language (strukturiran poizvedovalni jezik)

XLIFF – XML Localization Interchange file format (XML format za izmenjavo lokalizacijskih datotek)

ZVOP – Zakon o varstvu osebnih podatkov

ZinfV – Zakon o informacijski varnosti

ZZavar – Zakon o zavarovalništvu

1 UVOD

Razvoj informacijskih rešitev v sodobnem digitalnem svetu poteka z izjemno hitrostjo in vsakodnevnimi novostmi. Organizacije so zaradi tega pod velikim pritiskom, da razvijajo informacijske rešitve hitro, učinkovito in z minimalnimi stroški. Za razvoj informacijskih rešitev se zato vedno več podjetij odloča za malokodna razvojna okolja, ki omogočajo hitro izdelavo s pomočjo vizualnih gradnikov in predpripravljenimi komponentami. Ob tem pri digitalni transformaciji igra vedno večjo vlogo tudi umetna inteligenca (angl. AI – *Artificial intelligence*).

1.1 Opis področja in opredelitev problema

Področje raziskovanja v tem diplomskem delu zajema razvoj sodobnih informacijskih rešitev v malokodnem razvojnem okolju Oracle Apex. Oracle Apex je platforma, ki omogoča vizualno podprt razvoj informacijskih rešitev. Poseben poudarek je namenjen IAM (angl. *identity and access management*) in implementaciji revizijskih sledi, ki omogočajo pregled nad poslovnimi dogodki. Prav tako so raziskane generativne AI rešitve, ki s pomočjo naprednih jezikovnih modelov ustvarjajo odgovore na podlagi vhodnih podatkov.

Raziskovalni problem je povezan z vprašanjem, kako učinkovito nadgraditi IAM rešitev z revizijskimi sledmi znotraj malokodnega razvojnega okolja Oracle Apex. Preveriti smo želeli, ali ta pristop omogoča izdelavo kompleksnih aplikacij brez potrebe po programskem znanju. Prav tako nas zanima, ali uporaba IAM rešitve zagotavlja preglednost in kontroliran dostop do funkcionalnosti v informacijskih rešitvah. Diplomsko delo zajema primerjavo različnih oblik hranjenja in beleženja sprememb v informacijskih rešitvah. Nazadnje se preverja še delovanje dveh generativnih AI storitev, OpenAI in Cohere, in ovrednotenje, katera storitev je učinkovitejša na primeru vgrajenega pogovornega vmesnika v malokodnem razvojnem okolju Oracle Apex.

1.2 Namen, cilji in osnovne trditve

Namen diplomskega dela je raziskati in predstaviti razvoj sodobnih informacijskih rešitev v malokodnem razvojnem okolju Oracle Apex in prikazati primer razširitve funkcionalnosti

obstoječe IAM centralne rešitve. Prikazan je tudi primer uporabe AI v procesu razvoja aplikativne rešitve.

Cilji diplomskega dela:

1. Opisati malokodno razvojno okolje Oracle Apex in predstaviti njegovo uporabo pri razvoju sodobnih informacijskih rešitev.
2. Predstaviti sistem za upravljanje dostopov (IAM) in njegovo vlogo v sodobnih informacijskih rešitvah.
3. Razložiti pojem in pomen revizije in revizijskih sledi ter prikazati implementacijo v obstoječo poslovno rešitev.
4. Raziskati možnosti vključevanja AI v proces razvoja in nadgradnje funkcionalnosti aplikativne rešitve.
5. Implementirati beleženje revizijske sledi v obstoječo IAM rešitev.
6. Implementirati AI pomočnika za generiranje poizvedb in prikaz rezultatov.

Hipoteze diplomskega dela:

H1: Izdelava poslovnih informacijskih rešitev v malokodnem razvojnem okolju Oracle Apex ne zahteva osnovnega znanja o programiranju.

H2: Uporaba IAM rešitve zagotavlja transparenten in kontroliran dostop do funkcionalnosti v informacijskih rešitvah.

H3: Hranjenje zgodovinskih podatkov v podatkovnih tabelah ne omogoča revizijskega vpogleda v izvajanje poslovnih procesov znotraj informacijske rešitve.

H4: Generativna AI rešitev OpenAI storitev je učinkovitejša v primerjavi s Cohere storitvijo v malokodnem razvojnem okolju Oracle Apex na primeru vgrajenega pogovornega vmesnika.

1.3 Predpostavke in omejitve

Obstoječa informacijska rešitev, ki je obravnavana v diplomskem delu, temelji na že narejenem sistemu, ki je v uporabi že dlje časa in jo uporablja več različnih skupin uporabnikov z različnimi dostopi ter načini uporabe. To vpliva na njeno strukturo in morebitne omejitve pri nadgradnji. Dostop do podatkov pri nadgradnji je omejen le na izbrane tabele in attribute. Zaradi varovanja osebnih podatkov in podatkov podjetja so vsi podatki anonimizirani ali psevdonimizirani. Nadgradnja je izvedena znotraj razvojne sheme in ni neposredno vključena

v produkcijsko okolje. S tem želimo zagotoviti varnost, stabilnost in neodvisnost od obstoječih delovnih procesov.

Pri razvoju se uporabljajo razvojna orodja podjetja Oracle. To predstavlja omejitve z vidika širšega primerjalnega vrednotenja z drugimi tehnologijami. Prav tako se uporabljata dve AI storitvi, OpenAI in Cohere, kar ponovno predstavlja omejitev z vidika primerjalnega vrednotenja.

Dodatna omejitev izhaja iz razpoložljivosti literature. Zaradi hitrega razvoja tehnologije je težko dostopati do aktualnih znanstvenih virov, ki bi bili v celoti usklajeni s trenutnim stanjem na področju. Najnovejši podatki so večinoma zajeti v tehnični dokumentaciji in publikacijah podjetij, ki se neposredno ukvarjajo z razvojem teh rešitev. Večina virov torej temelji na tuji literaturi, člankih in povzetkih s konferenc.

1.4 Uporabljene raziskovalne metode

V diplomskem delu se uporabljajo različne raziskovalne metode, tako v teoretičnem kot praktičnem delu. Uporablja se deskriptivna metoda oziroma opisna analiza. S pomočjo te metode se obravnava relevantna strokovna in tehnična literatura s področja teme diplomskega dela. S to metodo je oblikovana teoretična zasnova, ki služi kot podlaga praktičnemu delu diplomskega dela.

Uporablja se razvojno-aplikativna metoda, saj se v diplomskem delu na konkretnem primeru izvaja nadgradnja obstoječe informacijske rešitve, ki je podrobno opisana. Opisani so uporabljeni postopki, orodja in morebitne omejitve, težave in rešitve, ki so nastale tekom izvajanja nadgradnje. Ob tem je uporabljena študija primera, s katero je izvedena globoka analiza in vrednotenje učinkovitosti izvedene nadgradnje.

Uporablja se tudi empirična kvantitativna metoda, saj numerično vrednotimo odgovore AI pomočnika in jih nato statistično analiziramo, primerjamo in interpretiramo.

2 MALOKODNA RAZVOJNA OKOLJA

Malokodna razvojna okolja (angl. *low-code platforms*) omogočajo razvoj informacijskih rešitev z uporabo vizualnih orodij in minimalnim programiranjem. Z vmesniki tipa "*point and click*" uporabnik sestavi funkcionalnosti, kot so regije različnih oblik, obrazci, gumbi ... Tak pristop omogoča hitrejši razvoj rešitev z manj potrebnega tehničnega znanja, kar predstavlja dostopnejšo alternativo klasičnemu programiranju (What is low-code, 2025).

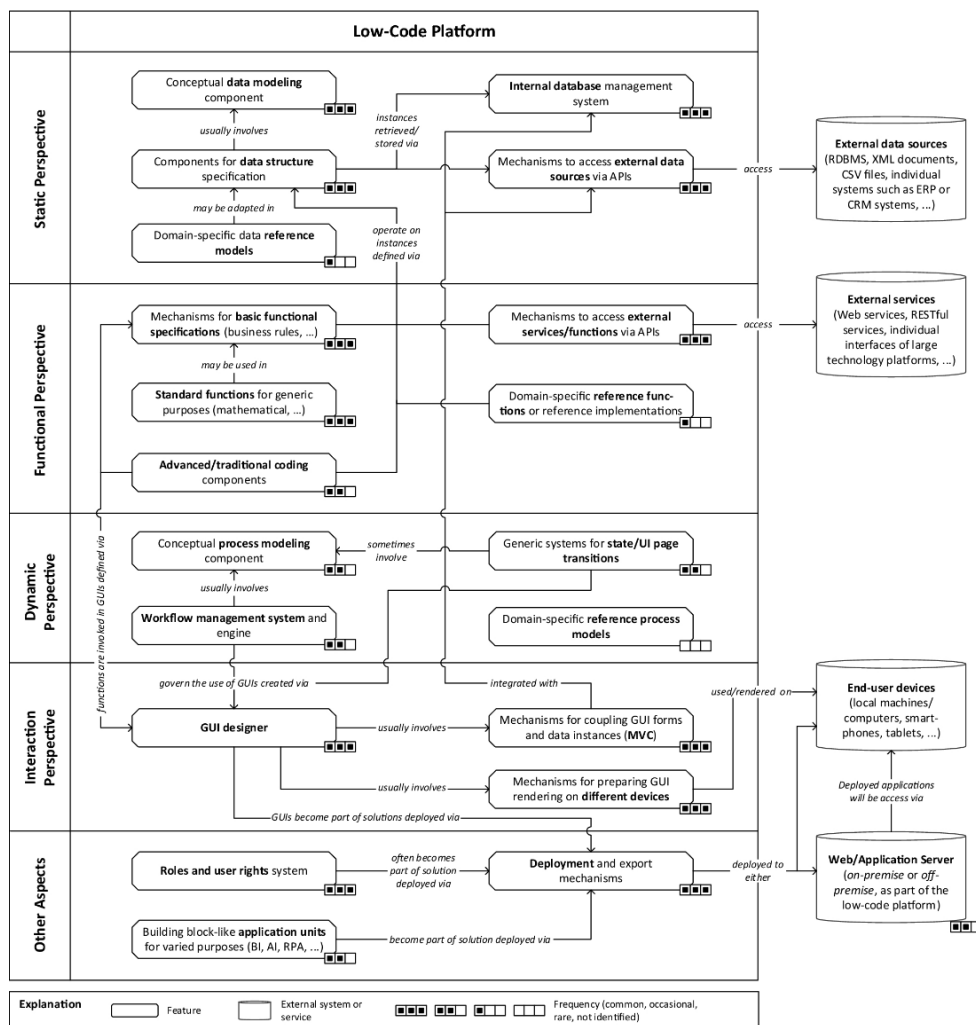
Sistemi v malokodnih razvojnih okoljih so intuitivni in uporabniku prijazni. Namesto obsežne kode se gradijo rešitve, ki omogočajo dostop do podatkovnih baz, izvajanje analiz in komunikacijo s poslovnimi sistemi. Vedno več razvojnih okolij prav tako podpira uporabo AI (prav tam).

Opremljenih je pet glavnih kategorij malokodnih razvojnih okolij, in sicer generalna uporaba, procesne aplikacije, podatkovne aplikacije, sistemi za upravljanje zahtevkov in mobilne aplikacije. V tem diplomskem delu se bomo osredotočili na podatkovne aplikacije (Richardson & Rymer, 2016).

2.1 Značilnosti in namen

Malokodna razvojna okolja za podatkovne aplikacije omogočajo zbiranje, obdelavo in prikaz podatkov iz relacijskih podatkovnih baz. Osnovna uporaba malokodnih razvojnih okolij je možna brez pisanja kode. Za pisanje kode so potrebna različna znanja o programskih jezikih in tehnikah. Znanja jezikov se nanašajo na programske jezike, kot so CSS, JavaScript, HTML, SQL ... Tehnike pa se nanašajo na znanje o algoritmičnih in strukturah. Malokodna razvojna okolja omogočajo implementacijo skript in prilagoditve z uporabo različnih programskih jezikov. Glavni ponudniki malokodnih razvojnih okolij so Oracle, Alpha software, Caspio, Magic Software (prav tam). V diplomskem delu se bomo osredotočili na Oracle.

Malokodna razvojna okolja se uporabljajo za različne namene, kot je izboljšanje uporabniške izkušnje, za poslovne informacijske rešitve in aplikacije za boljšo učinkovitost (What is low-code, 2025). Raziskava Bock in Frank (2021) povzema, da večina malokodnih razvojnih okolij ponuja podporo za različne funkcionalnosti, ki so razdeljene na področja: statična, funkcionalna, dinamična, interaktivna in drugo. Na sliki 1 je prikazano, kako pogosto se funkcionalnosti pojavljajo na platformah.



Slika 1: Malokodne platforme

Vir: (Bock & Frank, 2021)

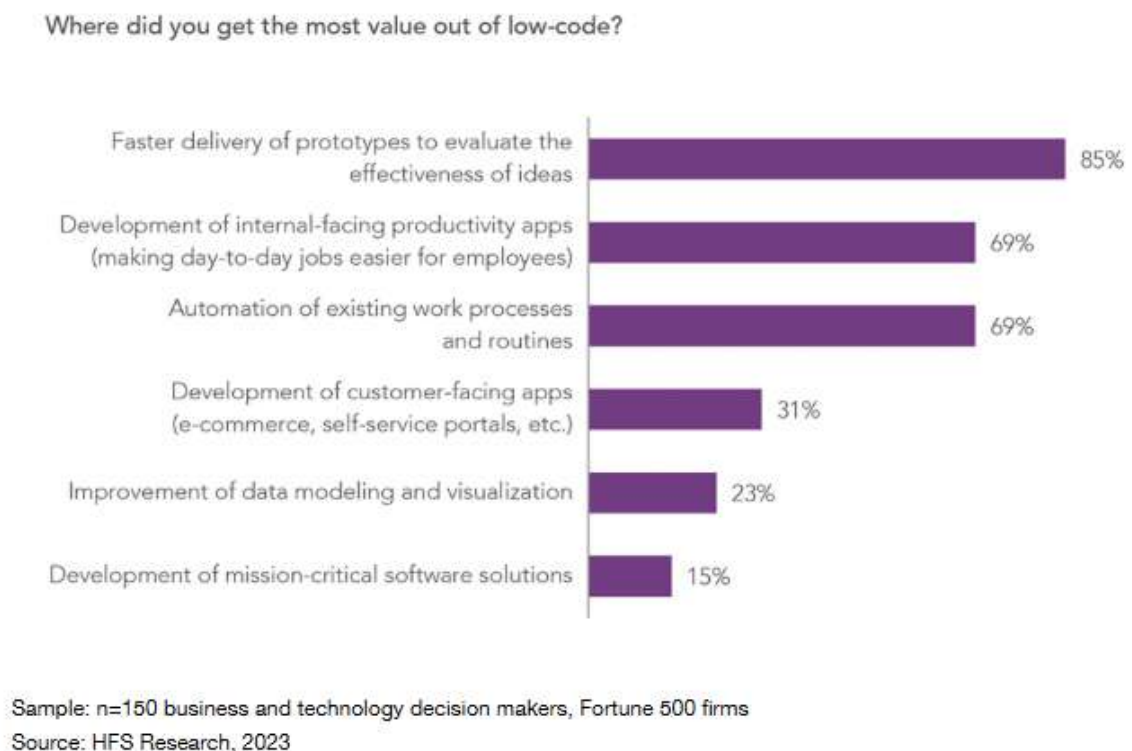
2.2 Prednosti

Prednost malokodnih razvojnih okolij je prilagodljivost, saj omogočajo izdelavo informacijskih rešitev z uporabo vizualnih orodij, ob tem pa dopuščajo vnos lastne kode v različnih jezikih. Razvoj je hiter, kar podjetjem prihrani čas in zmanjša stroške. Zaradi enotnih predlog in vmesnikov so rešitve dosledne in enostavne za uporabo ter vzdrževanje. Tovrstna razvojna okolja vzpodbujajo sodelovanje znotraj organizacij, hkrati pa podpirajo agilnost, saj omogočajo hitro prilagajanje potrebam trga (What is low-code, 2025).

Malokodna razvojna okolja se razvijajo iz četrte generacije programskih jezikov (angl. 4GL – *fourth generation languages*), katerih glavne značilnosti so deklarativnost, strukturiranost, manjša uporaba programske kode in poudarek na poslovni logiki. Presegajo jih v večji zmogljivosti in prilagodljivosti. Malokodna razvojna okolja namreč pogosto vključujejo

programske vmesnike za aplikacije (angl. API – *application programming interface*), omogočajo izvoz kode in povezovanje z drugimi sistemi. To povečuje njihovo odpornost in interoperabilnost. Ob razvoju informacijskih rešitev prav tako omogočajo oziroma vključujejo orodja za nameščanje, upravljanje okolij in podporo celotnemu življenjskemu ciklu aplikacij (Richardson & Rymer, 2016).

Slika 2 prikazuje raziskavo, kjer so 150 podjetij vprašali (*Fortune 500 firms*), kje vidijo največje prednosti uporabe malokodnih razvojnih okolij (Martin, 2023).



Slika 2: Mnenja podjetij Fortune 500 firms

Vir: (Martin, 2023)

2.3 *Razvoj in sodobni trendi*

V prihodnosti bodo malokodna razvojna okolja nudila še večjo podporo povezovanja z zunanjimi podatkovnimi viri in API, hkrati pa bodo nadgrajena z naprednimi orodji za povezovanje s podatkovnimi bazami in uporabo čarovnikov za razvoj. Pri tem bo veliko vlogo igrala AI. Prav tako se v prihodnosti pričakuje več sodelovanj, tako v samem razvojnem okolju kot z drugimi okolji. Okolja bodo prav tako postajala vse bolj skalabilna in primerna za nameščanje v distribuirana strežniška okolja (What is low-code, 2025).

N. Prinz, Rentrop in M. Huber (2021) opozarjajo, da se dosedanje raziskave o malokodnih razvojnih okoljih osredotočajo na tehnične vidike, zanemarljivo pa se organizacijske in človeške dejavnike. Predlagajo, da podjetja pri uvedbi teh okolij ne upoštevajo samo tehnični vidik, temveč tudi človeški vidik (odnosi, veščine in organizacijsko kulturo podjetja).

2.3.1 *Podpora umetne inteligence*

AI se nanaša na računalniške sisteme, ki so zmožni opravljati naloge, za katere se sicer domneva, da zahtevajo človeško inteligenco. To so na primer pisanje programov, reševanje matematičnih problemov, razumevanje jezika, načrtovanje in sklepanje na podlagi kontekstualnega znanja (Nilsson, 1980). V tem poglavju bomo predstavili, kakšno podporo ponuja AI v malokodnih razvojnih okoljih.

AI dodatno nadgrajuje zmogljivosti malokodnih razvojnih okolij. S pomočjo strojnega učenja in procesiranja naravnega jezika (angl. NLP – *natural language processing*) lahko uporabniki opišejo željeno funkcionalnost, AI pa ponudi ustrezen odgovor in/ali kodo. Poleg tega AI omogoča analizo, optimizacijo in odpravljanje napak v obstoječih rešitvah. AI prav tako prevzema rutinska opravila, kar razvijalcem omogoča, da se osredotočijo na strateške naloge, varnost in inovacije. Omeniti je pomembno še pogovorne sisteme, ki omogočajo poizvedbe v naravnem jeziku brez potrebe po navajanju v strukturirani obliki (Alionco, 2024).

Kljub prednostim pa je potrebno opozoriti na težave, ki se lahko pojavijo z uporabo AI. AI lahko predstavlja potencialne ranljivosti in zlorabe, zato je nujen nadzor nad delovanjem sistemov ter jasna delitev odgovornosti med AI in razvijalci. Posebna pozornost mora biti usmerjena na zaščito podatkov in s tem sklenitev jasnih dogovorov s ponudniki AI storitev glede obdelave in hrambe podatkov. V poslovnem okolju je ključna tudi etika rabe AI.

Uporabnike je treba namreč obvestiti, če se v razvojnem okolju uporablja AI, kako se uporablja in zakaj (Alionco, 2024).

2.3.2 AI pogovorni sistemi

Pogovorni sistemi, ki jih poimenujemo tudi dialogni sistemi (angl. *chatbots*), so računalniški programi, s katerimi poteka komunikacija v naravnem jeziku. Delijo se na:

- Sistemi za informacijsko iskanje: odgovarjajo na specifična vprašanja, pogosto z uporabo baz znanja ali iskalnikov.
- Sistemi za upravljanje opravil: omogočajo izvajanje opravil, kot so nakupi/rezervacije.
- Sistemi za socialno interakcijo: naraven pogovor brez specifičnega cilja (Jurafsky & Martin, 2023).

Standardna arhitektura vključuje komponente: prepoznavanje naravnega jezika (angl. NLU – *natural language understanding*), upravljalnik dialoga in generiranje naravnega jezika (angl. NLG – *natural language generation*) (prav tam).

3 ORACLE APEX

Oracle Apex, krajše za *Oracle Application Express*, je malokodno razvojno okolje. S pomočjo Oracle Apex uporabniki razvijajo in uporabljajo informacijske rešitve preko spletnega vmesnika. Vsebuje orodja, s katerimi se lahko gradijo različne vrste informacijskih rešitev oziroma aplikacij, osnovanih na relacijskih podatkovnih modelih. To vključuje vnos, shranjevanje, prikazovanje, spreminjanje, izračunavanje, logične operacije nad podatki ... Z uporabo jezika SQL in PL/SQL, se ustvarijo uporabniški vmesniki, ki generirajo različna poročila in uporabljajo spletne informacijske rešitve (Leskovar & Baggia, 2024).

Oracle Apex opravlja dve osnovni nalogi, in sicer izris (angl. *rendering*) in obdelavo spletnih strani (angl. *processing*). Spletne strani se generirajo iz metapodatkov, ki jih uporabnik deklarativno določi v spletnem razvojnem vmesniku (Cimolini, 2017).

Oracle Apex danes uporablja že več kot 850 000 razvojniki. S pomočjo Oracle Apex je bilo zgrajenih več kot 21 milijonov aplikacij (Zeichick, 2024). Uporabljajo ga številna podjetja po svetu (Siemens, Vodafone, Samsonite, Munich RE, BMI Healthcare, Isa) (Success stories, 2025) kot tudi mnoga podjetja v Sloveniji.

3.1 Lastnosti

Oracle Apex je zasnovan na Oracle relacijski podatkovni bazi in programskem jeziku SQL in PL/SQL. Podatki za izdelavo informacijskih rešitev se lahko pridobijo iz lokalnih podatkovnih baz, oddaljenih podatkovnih baz ali spletnih storitev (Features, 2025).

Oracle Apex ponuja možnost generiranja različnih oblik izpisov podatkov, npr. preproste tabelarične oblike ali naprednejše, interaktivne preglednice in druge oblike vizualizacije podatkov. Oracle Apex podpira izdelavo progresivnih spletnih aplikacij (angl. PWA – *progressive web apps*), ki se lahko namestijo kot samostojne aplikacije na različnih napravah. Univerzalna tema, ki definira uporabniški vmesnik, je namreč zasnovana tako, da omogoča prilagodljivost aplikacij različnim velikostim zaslonov (angl. *responsive design*). S funkcijami "theme roller" in "template options" je mogoče dodatno prilagoditi videz aplikacij. Oracle Apex ponuja obsežno knjižnico ikon (več kot 11 000), ki jih je možno prilagoditi, lahko pa uporabimo tudi lastne. Za podporo globalni uporabi Oracle Apex omogoča večjezičnost preko XLIFF prevajalnika (angl. XML – *localization interchange file format*) in prilagajanje prikaza glede na jezik, lokacijo ali časovni pas uporabnika (prav tam).

Razvijalcem omogoča spremljanje delovanja aplikacij, vključno z načinom razhroščevanja, pregled SQL poizvedb in beleženjem aktivnosti, kot so prijave, ogledi strani in spremembe. Platforma vključuje tudi REST delavnico, ki podpira pridobivanje in manipulacijo podatkov preko oddaljenih virov (Features, 2025).

3.2 Zgradba in delovanje

Oracle Apex temelji na preprosti metapodatkovni arhitekturi, ki omogoča hiter dostop do podatkov, skalabilnost in dobro zmogljivost. Uporablja trinivojsko arhitekturo, kjer uporabniške zahteve iz brskalnika preko spletnega strežnika potujejo do podatkovne baze. Vsa obdelava, logika in manipulacija podatkov se izvede neposredno v podatkovni bazi (Architecture, 2025).



Slika 3: Diagram arhitekture Oracle Apex

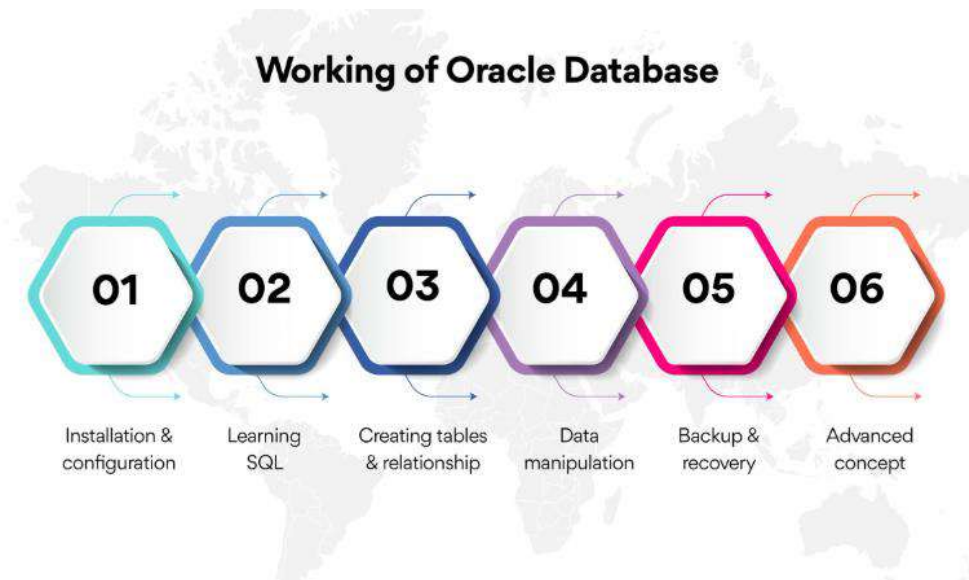
Vir: (Architecture, 2025)

Dodatno Oracle Apex generira spletne strani, ki z uporabo tehnologije AJAX (angl. *asynchronous JavaScript and XML*) omogoča počasno nalaganje (angl. *lazy loading*) vsebin in asinhrono osveževanje podatkov brez potrebe po ponovnem nalaganju strani. S tem se izboljša odzivnost aplikacij. Vgrajene validacije in RESTful storitve pa razbremenijo strežnik, medtem ko "Apex collection" omogoča začasno shranjevanje podatkov znotraj posamezne seje (Syed, 2024).

3.2.1 Oracle podatkovna baza in PL/SQL

Poznamo več definicij podatkovne baze. Podatkovna baza je model okolja, katere namen je sprejemanje odločitev in izvajanje akcij. Prav tako je deljena zbirka med seboj logično povezanih podatkov za informacijske potrebe poslovnega sistema. Podatki v bazi so shranjeni v računalniškem sistemu, do katerih je dostop centraliziran in omogočen s sistemom za upravljanje s podatkovnimi bazami. Imenuje se SUPB – sistem za upravljanje podatkovnih baz (angl. DBMS – *database management system*) (Mohorič, 1992).

Oracle podatkovna baza temelji na relacijskem podatkovnem modelu, ki je najpogostejše uporabljen model v poslovnih informacijskih rešitvah (Rožanec, 2017). Primerna je za vse vrste organizacij (male, srednje in velike). Omogoča umestitev v različne oblike arhitekturnih rešitev in namestitev na Windows, UNIX in Linux operacijske sisteme ali uporabo v oblaku. Glavne značilnosti so, da za varnost podatkov uporablja standardne algoritme za šifriranje podatkov, zaseda malo prostora, podpira hranjenje in zmogljivo obdelavo velike količine podatkov (Zorman, 2024). V odvisnosti od infrastrukture, velikosti organizacij in števila orodij so stroški za uporabo Oracle storitev različni. Za manjše projekte so stroški lahko nični ali zelo majhni, nato pa rastejo v odvisnosti z zgoraj podanimi kriteriji.



Slika 4: Oracle podatkovna baza

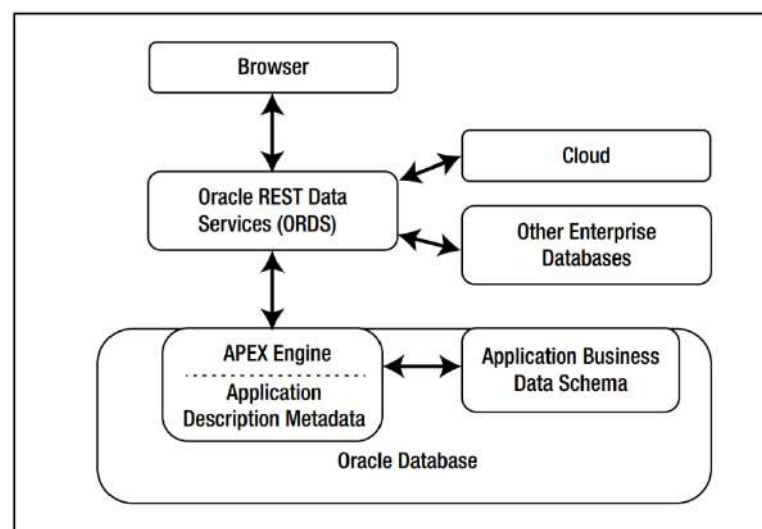
Vir: (Gillani, 2025)

Večina podatkovnih baz za dostop in obdelavo uporablja SQL (angl. *structured query language*), ki je bil razvit v 80. letih prejšnjega stoletja (What is a database?, 2020).

PL/SQL je proceduralni programski jezik. Omogoča neposredno vključevanje SQL stavkov in izvajanje programskega toka, uporabo parametrov in pisanje procedur ter funkcij. Njegova prednost je zmožnost hranjenja poslovne logike v sami podatkovni bazi in izvajanje znotraj aktivne transakcije (Database concepts, 2025).

3.2.2 Aplikacijski strežnik ORDS

ORDS (angl. *Oracle REST data services*) je spletni vmesnik, zgrajen na Java EE platformi (angl. *Java platform, enterprise edition*), namenjen objavi spletnih storitev, ki služijo povezovanju aplikacij z baznimi storitvi preko protokola HTTPS (angl. *HyperText transfer protocol secure*). Omogoča dostop do Oracle Apex funkcionalnosti preko spletnega vmesnika, proženje baznih procedur in funkcij ter izgradnjo REST API vmesnikov za dostop do podatkov (Introduction to ORDS, 2025).



Slika 5: ORDS

Vir: (Cimolini, 2017)

3.2.3 Večnivojska arhitektura

Na sliki 6 je prikaz primera večnivojske arhitekture. Vsebuje štiri nivoje: podatkovni nivo, nivo poslovne logike, storitveni nivo in predstavitveni nivo. Kot je razvidno s slike, se Oracle Apex uporablja v predstavitvenem nivoju.



Slika 6: Večnivojska arhitektura

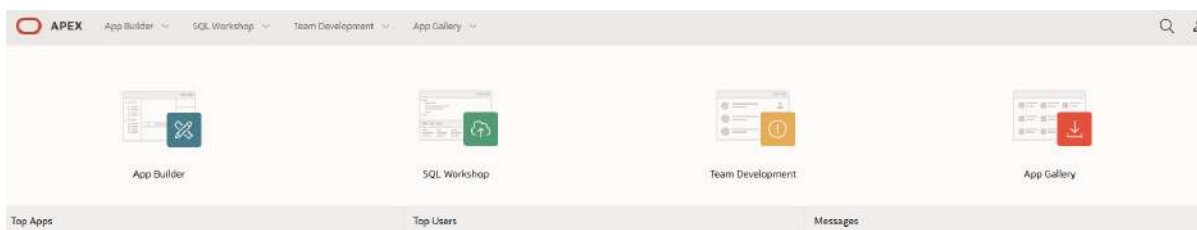
Vir: (Zorko, 2025)

3.3 Možnosti namestitve v različnih okoljih

Programska oprema se lahko gosti in uporablja na različne načine, in sicer kot programsko opremo v oblaku (angl. SaaS – *Software as a Service*), kot platformno storitev (angl. PaaS – *Platform as a Service*) ali kot samostojno nameščeno rešitev (angl. *self-hosted*) (Kauflig, 2021). Oracle oblaka infrastruktura (angl. OCI – *Oracle Cloud Infrastructure*) je prva avtonomna infrastruktura, ki vključuje vse potrebne storitve – od obdelave in shranjevanja podatkov do spletnih razvojnih orodij. Je predkonfigurirana, varna, visoko razpoložljiva in globalno dostopna (Apex oracle cloud, 2025). Oracle dodatno ponuja brezplačno različico Apex kot spletno aplikacijo na spletnem brskalniku (Leskovar & Baggia, 2024).

3.4 Upravljanje okolja

Oracle Apex sestoji iz enega ali več delovnih okolij (angl. *workspaces*). Orodja so razdeljena v štiri sklope: graditelj aplikacij (angl. *app builder*), SQL delavnica (angl. *SQL Workshop*), podpora razvoju v timu (angl. *team development*) in galerija aplikacij (angl. *app gallery*).

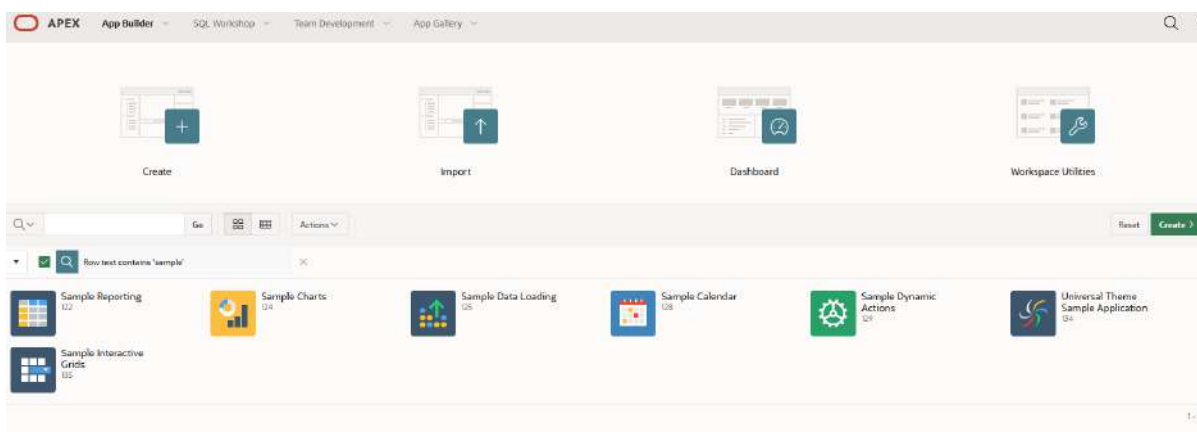


Slika 7: Oracle Apex domača stran

Vir: (Lastni vir)

Graditelj aplikacij je namenjen ustvarjanju in urejanju aplikacij. Apex aplikacije so spletni uporabniški vmesniki, ki omogočajo interaktivno izvajanje poslovnih procesov, ki temeljijo na obstoječem poslovnem podatkovnem modelu in poslovni logiki, lahko pa se tudi nadgrajujejo (Leskovar & Baggia, 2024).

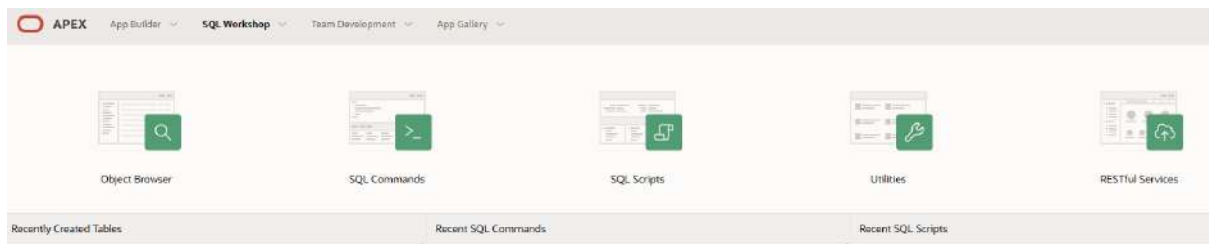
Kot je razvidno s slike 8, je stran sestavljena iz več gradnikov. Prikazane so ikone za ustvarjanje nove aplikacije, uvoz aplikacij, dostop do nadzorne plošče in orodij delovnega prostora. Za izbran delovni prostor so pod tem razvrščene posamezne aplikacije.



Slika 8: Oracle Apex graditelj aplikacij

Vir: (Lastni vir)

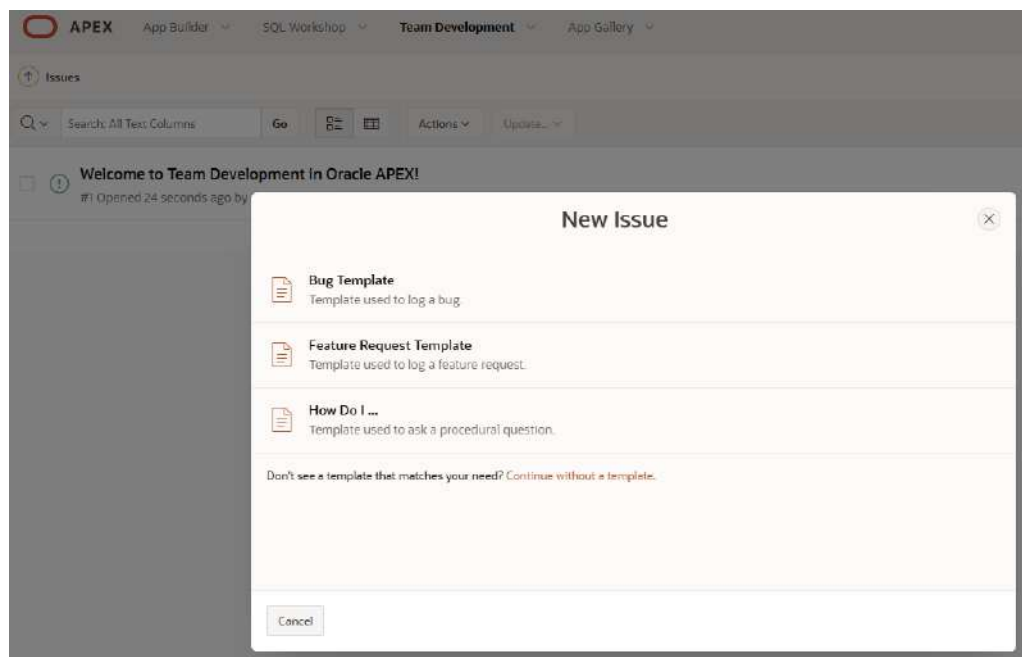
Naslednja pomembna komponenta je SQL delavnica (slika 9). V njej je mogoče iskati in pregledovati podatkovne objekte, pisati SQL ukaze, upravljati in izvajati skripte, uporabljati dodatna pomožna orodja in upravljati z REST spletnimi storitvami.



Slika 9: Oracle Apex delavnica SQL

Vir: (Lastni vir)

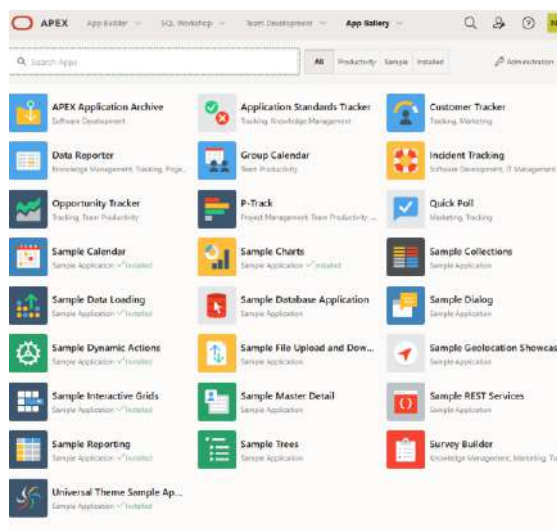
Sledi modul za podporo timskemu delu (slika 10). Namenjen je podpori sodelovanja znotraj ene razvojne ekipe. Omogoča pregled nalog, prijavljenih napak, vprašanj in drugih elementov.



Slika 10: Oracle Apex skupinsko razvoj

Vir: (Lastni vir)

V galeriji aplikacij (slika 11) so zbrane vzorčne oziroma predstavitvene aplikacije. Uporabljajo se lahko za predstavitev določenih funkcionalnosti ali pa se jih uporabi kot predlogo pri ustvarjanju lastnih aplikacij.

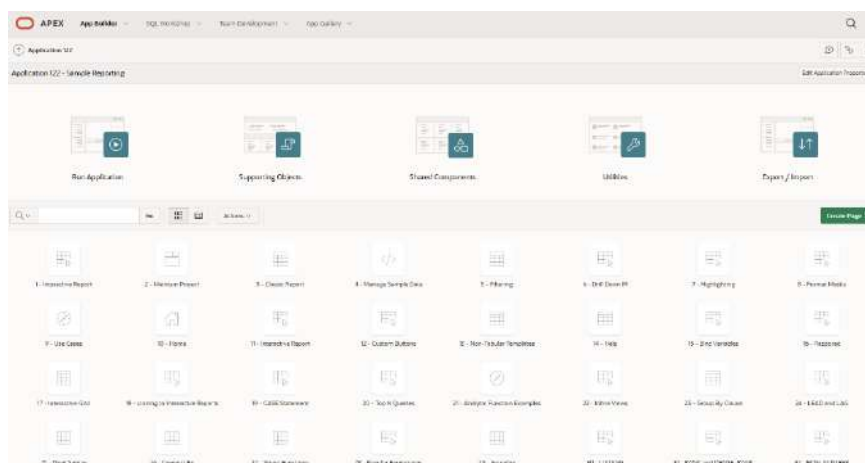


Slika 11: Oracle Apex galerija aplikacij

Vir: (Lastni vir)

3.5 Upravljanje aplikacij

Aplikacije se upravljajo v razdelku graditelj aplikacij (slika 12). V razdelku lahko izdelamo novo aplikacijo ali uporabimo že izdelane predloge. Vsaka aplikacija vsebuje več razdelkov. Razdelki so: zaženi aplikacijo, povezani objekti, skupne komponente, orodja in izvoz/uvoz.



Slika 12: Oracle Apex graditelj aplikacij

Vir: (Lastni vir)

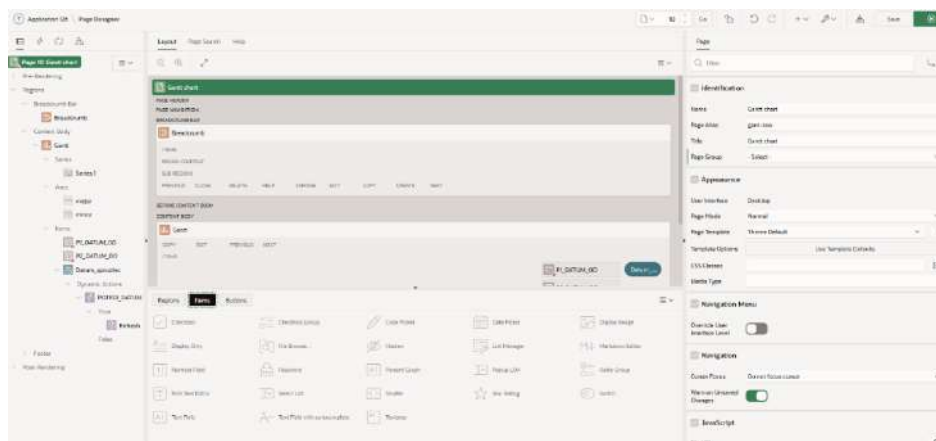
Novo stran ustvarimo s klikom na gumb "*ustvari stran*" (angl. *create page*). Če kliknemo na eno izmed že narejenih strani, pa se prikaže oblikovalnik strani.

Strani lahko uporabljajo skupno temo in predlogo, vendar jih je možno tudi individualno prilagoditi. Na spodnjem posnetku zaslona je prikazan izgled oblikovalnika strani na primeru strani s prikazom Ganttovega diagrama.

Kot je razvidno s slike 13, je oblikovalnik strani sestavljen iz treh stolpcev.

- Levi stolpec prikazuje regije strani, elemente in gumbе, razporejene po sekvenci oziroma vrstnem redu prikaza na strani od zgoraj navzdol.
- Srednji stolpec prikazuje postavitev elementov na strani - izgled strani.
- Desni stolpec vsebuje nastavitve, ki jih je možno prilagajati za celotno stran ali za vsak posamezen element.

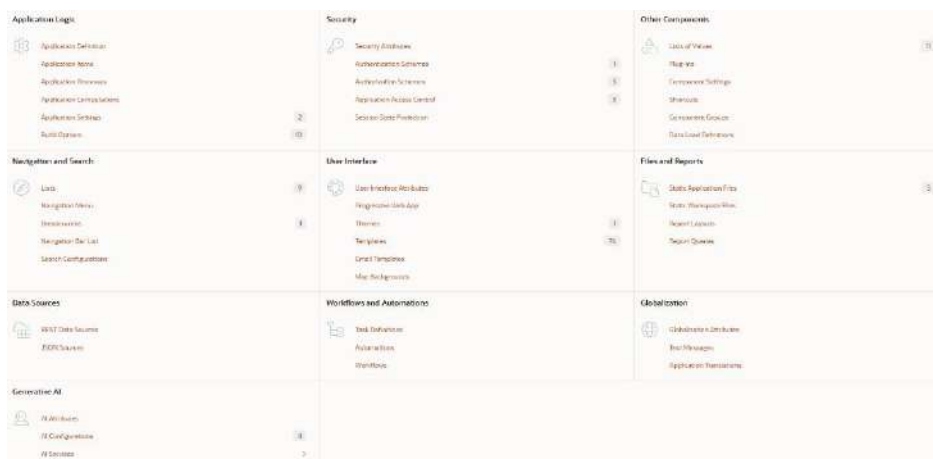
V zgornjem desnem kotu pa se nahajajo še gumbi za zagon aplikacije, shranjevanje sprememb in dostop do skupnih komponent.



Slika 13: Oracle Apex oblikovalnik strani

Vir: (Lastni vir)

Skupne komponente so gradniki, ki jih lahko uporablja več strani ali več aplikacij znotraj istega delovnega okolja. Kot je razvidno s slike 14, so ključne kategorije: aplikacijska logika, navigacija, viri podatkov, uporabniški vmesnik, poročila, dodatne komponente, datoteke in globalizacija. V različici 24.1 se je h komponentam vključila nova kategorija, ki se imenuje generativna AI.



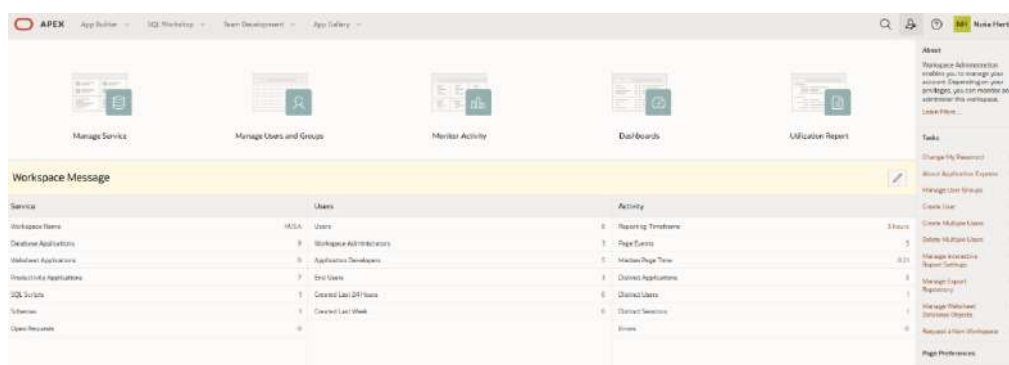
Slika 14: Oracle Apex skupne komponente

Vir: (Lastni vir)

3.5.1 Administracija

Gradniki administracijskega vmesnika so domača stran, upravljanje storitev in upravljanje delovnega okolja.

Administracijska domača stran omogoča upravljanje petih gradnikov, ki so upravljanje storitev, upravljanje uporabnikov in skupin, nadzor aktivnosti, nadzorna plošča in poročilo o izkoriščenosti (angl. *utilization report*). Prav tako vključuje osnovne lastnosti posameznega delovnega okolja, kot je prikazano na sliki 15.



Slika 15: Oracle Apex administracija

Vir: (Lastni vir)

3.6 Rešitve z uporabo umetne inteligence

Podpora AI je bila v Oracle Apex prvič integrirana v različici 24.1. AI temelji na generativni inteligenci in je namenjen pomoči pri razvoju aplikacij (AI in Oracle APEX 24.1., 2025). Apex AI pomočnik nudi pomoč pri sestavljanju SQL poizvedb z uporabo naravnega jezika, avtomatizira pisanje SQL poizvedb in razloži obstoječo kodo ter predlaga popravke napak. Čarovnik za ustvarjanje aplikacij omogoča ustvarjanje nove aplikacije z uporabo naravnega jezika. S tem se zgradi osnovna struktura aplikacije, razvijalec pa se lahko takoj posveti ustvarjanju bolj specifičnih funkcionalnosti (Mohindroo, 2024).

Pogovorni AI vmesnik omogoča končnim uporabnikom interakcijo z aplikacijo v naravnem jeziku preko predpripravljenega pogovornega okna (prav tam). Po predhodni konfiguraciji ponudnika umetne inteligence je AI možno uporabiti kot deklarativno ali programsko funkcionalnost. Na primer: uporabi se lahko dinamično dejanje, kjer se AI pomočnik prikaže v pojavnem oknu ali neposredno na strani aplikacije (AI in Oracle APEX 24.1., 2025). Generativno AI storitev ponujajo naslednji ponudniki: OpenAI, OCI in Cohere.

V različici 24.2 so bile funkcionalnosti dodatno razširjene. Dodan je bil RAG (angl. *retrieval-augmented generation*), nadgraditev podatkovnega vira storitev AI. Omogoča oblikovanje poizvedb v naravnem jeziku znotraj lastnih podatkovnih virov. Pri tem je potrebno konfigurirati sistemski pozdrav, vir podatkov (angl. *RAG source*) in nabor v naprej določenih dejanj, navodil, vsebin, prevodov, ki jih naj AI upošteva (angl. *system prompts*) (AI in Oracle APEX 24.2., 2025).



Slika 16: RAG v Oracle Apex

Vir: (<https://www.youtube.com/watch?v=tQgOje9XCkw>, 2025)

4 IAM

Upravljanje identitet in dostopov IAM (angl. *identity and access management*) je disciplina kibernetске varnosti, ki omogoča nadzorovan dostop uporabnikov do digitalnih vsebin in kako lahko uporabniki te vsebine upravljajo (Kosinski, 2024). IAM je torej rešitev, ki v informacijskem sistemu nekega podjetja nadzira in omejuje uporabnikom dostop do njegovih kritičnih virov (Sweeney, 2024).

4.1 Opis IAM rešitev

Glavni koncepti IAM so identifikacija, avtentikacija, avtorizacija in beleženje revizijskih sledi. Identifikacija uporabniku informacijske rešitve priredi njegovo unikatno identiteto. Identiteto lahko predstavlja preprosto uporabniško ime, sestavljeno iz imena in priimka uporabnika, enolična uporabniška šifra ali kompleksnejši zapis, ki predstavlja identiteto uporabnika le znotraj omejenih, povezanih informacijskih rešitev. Identifikacija je proces preverjanja obstoja in veljavnosti takšne identitete. Avtentikacija je postopek preverjanja istovetnosti uporabnika. Vključuje lahko različne oblike preverjanja: vnos in primerjava gesla, dešifriranje identitete s ključem na pametni kartici, primerjavo biometričnih odčitkov ... Avtorizacija je proces, ki uporabniku na osnovi dodeljenih vlog ali pravic omogoča dostop do omejenih virov ali funkcionalnosti. Revizijske sledi zagotavljajo sledljivost izvajanja vseh opisanih procesov, ki jih IAM sistem podpira (Wong, 2024).

Z naraščajočo uporabo oblačnih storitev in distribuiranih okolij postajajo rešitve IAM vse bolj pomembne. IAM zagotavlja, da imajo do občutljivih podatkov dostop le pooblaščenе osebe. Zagotavlja zaščito pred zunanjimi vdori, notranjimi zlorabami in prispeva k preprečevanju varnostnih incidentov. Poleg varnostnih bonitet ima IAM pomembno vlogo pri zagotavljanju zakonske skladnosti (prav tam).

Z uporabo IAM se odpravi veliko ročnih postopkov, kar poveča učinkovitost in zmanjša možnost človeških napak. Sistem omogoča dodeljevanje, prilagajanje in preklic dostopnih pravic glede na vlogo uporabnika, njegovo delovno mesto in odgovornosti. V IAM je prav tako vedno bolj vpletena analiza vedenja uporabnikov in AI. V nadaljnjem razvoju tehnologije se predvideva uporaba AI za odkrivanje sumljivih dejavnosti (angl. ITDR – *identity threat detection and response*) (Sweeney, 2024).

4.2 Osnovne funkcionalne zahteve

IAM sistemi so kompleksni, prilagodljivi in razširljivi, saj se prilagajajo potrebam posameznih podjetij. Vsebujejo različne funkcionalnosti, ki se uporabljajo glede na konkretne zahteve (Wong, 2024). Ključne funkcionalnosti IAM rešitev so upravljanje pravic dostopa za vlogo in lokacijo uporabnika, avtomatizacijo postopkov za preverjanje in odobritev dostopov ter preprečevanje kopičenja prekomernih pravic. Omogočajo tudi enotno prijavo in celovit nadzor nad dostopi. IAM rešitve morajo beležiti podatke o prijavah, imeti nadzor nad dodeljevanjem in odvzemu pravic, služijo lahko kot centralizirana shramba identitet ali pa se z obstoječimi ponudniki identitet povezujejo (Sweeney, 2024).



Slika 17: IAM funkcionalnosti

Vir: (Medhi, 2025)

Shramba uporabniških profilov (angl. *user repository*) vsebuje identitete in podatke o uporabnikih znotraj podjetja. Ni samo podatkovna baza, ampak je tudi strukturiran in dostopni način organiziranja, kateri osebi pripada katera identiteta. Upravljanje uporabniških pravic vključuje njihovo prilagajanje ob spremembah delovnega mesta, napredovanja ali prekinitve zaposlitve. Tako se zagotavlja ustrezna raven dostopa (Wong, 2024).

Enotna prijava (angl. SSO – *single sign on*) uporabniku omogoča dostop do različnih sistemov z eno samo prijavo, kar poenostavi uporabo in poveča varnost. Federacija identitet (angl. *identity federation*) koncept enotne prijave dodatno razširi, saj omogoča prepoznavo uporabnika v zunanjih sistemih (prav tam).

Arhitektura IAM rešitve je lahko zgrajena na različne načine. Ena izmed najpogostejših je AAA (angl. *authentication, authorization and accounting framework*), ki omogoča vzpostavitev strukture, nadzora dostopa in sledenje. Drugi pristop je upravljanje identitet in nadzora IGA (angl. *identity governance and administration*), ki vključuje zagotovitev pravilne namestitve, nadzor in izvajanje revizijskih sledi. Prav tako avtor priporoča upravljanje digitalnih identitet naprav in aplikacij (angl. *machine identity management*), kot so strežniki, stacionarne in mobilne naprave, programski vmesniki (API) (Sweeney, 2024).

Večji ponudniki IAM rešitev so Oracle, Microsoft, IBM, Okta, OneLogin (prav tam).

4.3 Avtentikacija in avtorizacija

Avtentikacijske storitve so ključnega pomena, saj omogočajo preverjanje, kdo lahko dostopa do informacijske rešitve. Obstajajo različne oblike avtentikacije. Po Wong (2024) in MiniOrange (2025):

- Multifaktorska avtentikacija (MFA) – uporaba več korakov preverjanja.
- Kontekstna avtentifikacija (angl. *step-up*) – se aktivira v naprej določenih situacijah, ki zahtevajo dodatno avtentikacijo, saj gre za občutljive podatke.
- RBAC (angl. *role based access control*) – dodeljuje dostop na podlagi vloge uporabnika v procesu, ki ga informacijska rešitev podpira.
- RBA (angl. *risk-based authentication*) – dinamično nastavljanje dostopa glede na faktorje, kot so lokacija, naprava, IP naslov, uporabniški vzorci ali čas dostopa.

Oracle Apex ponuja lastno avtentikacijsko shemo, ki jo je možno uporabiti neposredno ali povezati z zunanjo infrastrukturo, kot so oblačni ponudniki, podjetniški imeniki ali lokalni imenik znotraj Oracle Apex (Features, 2025).

Avtorizacijske sheme v Oracle Apex omogočajo natančen nadzor dostopa do aplikacije, posameznih strani ali njihovih komponent. Sheme je možno prilagoditi in uporabiti selektivno. Prilagajajo se lahko z uporabo PL/SQL (prav tam), kar bo predstavljeno v naslednjem poglavju.

4.3.1 Avtorizacijske sheme

Oracle Apex omogoča ustvarjanje različnih avtorizacijskih shem. Ponuja več tipov shem in hkrati njihovo dodatno prilagoditev. Posamezno avtorizacijsko shemo je možno povezati s posameznimi aplikacijskimi elementi, kot so strani, regije, vnosna polja, preglednice, gumbi.

Na sliki 18 je prikazano okno za ustvarjanje nove avtorizacijske sheme. Shema je povezana z aplikacijo, med aplikacije pa jih lahko prenašamo s kopiranjem. Za ustvarjanje sheme je potrebno navesti ime, tip sheme in sporočilo, ki se prikaže, ko uporabnik nima dostopa.

Slika 18: Čarovnik za avtorizacijsko shemo

Vir: (Lastni vir)

V Oracle Apex je možno izbirati med 11 tipi shem, opisanih na sliki 19.

Slika 19: Oracle Apex tipi avtorizacijskih shem

Vir: (Lastni vir)

5 REVIZIJSKE SLEDI

Kot je napisano v 4. členu ZInfV-1: “*Revizijska sled je nespremenljiva sled oziroma niz podatkov, ki se je zgodil v informacijskem sistemu ali napravi, z natančnim časovnim zapisom v obliki dnevniškega zapisa, ki omogoča natančen pregled vseh zapisov, povezanih z vsemi dogodki in vsemi shranjenimi informacijami, od nastanka podatka ali informacije naprej do trenutnega stanja.*” (Zakon o informacijski varnosti /ZInfV-1/, 2025).

Revizijska sled (angl. *audit trail*) je kronološki zapis dejavnosti v informacijskem sistemu, ki se uporablja za zagotavljanje varnosti, preglednosti, notranjih revizij in skladnosti z zakonodajo. Revizijske sledi so ključne pri odkrivanju zunanjih napadov in notranjih zlorab. Da je revizija učinkovita, morajo biti revizijski mehanizmi avtomatizirani in neizbrisljivi. S tem se ohrani celovitost in zanesljivost zabeleženih podatkov (Vicente, 2024).

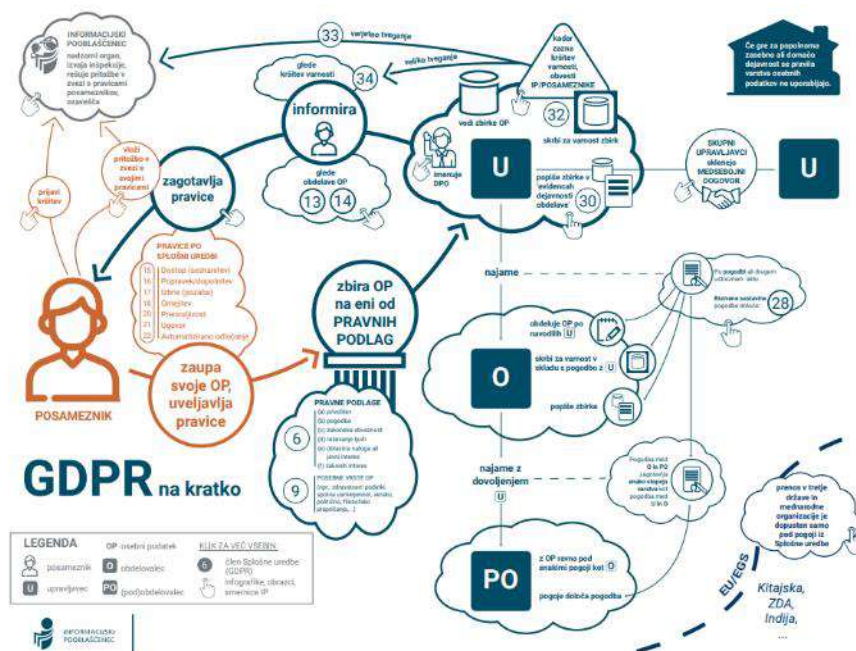
5.1 Zahteve

Najpogostejše zahteve za vodenje revizijskih sledi so regulatorne, še posebej tiste, ki se nanašajo na vodenje revizijskih sledi s področja varstva osebnih podatkov. Na tem področju obstaja več standardov, med katerimi se na tržišču najpogosteje pojavljajo sledeči:

- ISO/IEC (angl. *International Organization for Standardization/International Electrotechnical Commission*) 29100 je ogrodje, ki opredeljuje osnovne terminologije zasebnosti in opredeljuje vloge organizacijam v zvezi z zasebnostjo in varovanja le-te.
- ISO/IEC 27001 je standard za upravljanje informacijske varnosti.
- ISO/IEC 27701 je nadgradnja in razširjena opcija ISO/IEC 27001 in 27002. V ospredju je skrb za zaščito osebnih podatkov (Vincekovič, 2024).

GDPR (angl. *general data protection regulation*) je zakon Evropske unije o varovanju osebnih pravic, ki vključno z ZVOP-2 določa pravila o varovanju osebnih podatkov (prav tam).

ZVOP-2 je zakon o varstvu osebnih podatkov in je prišel v veljavo 26. 1. 2023. Zakon vsebuje številne določbe glede varnosti osebnih podatkov, vključno z zahtevami, ki jih je potrebno upoštevati pri vodenju revizijskih sledi oziroma dnevnika obdelave (Zakon o varstvu osebnih podatkov /ZVOP-2/, 2022).



Slika 20: GDPR na kratko

Vir: (Infografike, 2025)

Izvajalci bistvenih storitev morajo za obvladovanje incidentov hraniti dnevniške zapise o delovanju ključnih informacijskih sistemov najmanj šest mesecev. Hranjenje mora potekati na ozemlju Republike Slovenije, razen pri digitalni infrastrukturi, bančništvu in finančnem trgu, kjer je dovoljeno hranjenje tudi znotraj Evropske unije (Zakon o informacijski varnosti /ZInfV-1/, 2025).

Zavarovalnice in Slovensko zavarovalno združenje mora obdelovati osebne podatke v skladu z zakonodajo, ki ureja varstvo osebnih podatkov in zbirke podatkov s področja zavarovanja (Zakon o zavarovalništvu /ZZavar-1/, 2015).

5.2 Struktura zapisa

Dnevnik obdelave mora voditi zapise, kadar se obdelujejo posebne vrste osebnih podatkov ali kadar se z oceno tveganje določi, da je to potrebno (Zakon o varstvu osebnih podatkov /ZVOP-2/, 2022).

Revizijska sled oziroma evidenca dejavnosti obdelav mora vsebovati naslednje podatke: vrsto dejanja obdelave, datum in čas obdelave, identifikacijo osebe, ki je izvedla dejanje obdelave in identifikacijo uporabnikov osebnih podatkov. Dodatne vsebine lahko določi upravljalec (prav tam).

Po Vicente (2024) revizijski zapis vključuje uporabniško identiteto (uporabniško ime/ID), dogodek (prijava, dodajanje, spreminjanje, brisanje), datum in čas dogodka, vir oziroma kraj dostopa (IP naslov, naprava), rezultat dejanja (uspešno/neuspešno). Informacije so načeloma shranjene v strukturirani obliki – tabeli.

5.3 *Vpliv beleženja na delovanje aplikacij*

Beleženje revizijskih sledi lahko vpliva na zmogljivost sistema. Obsežno beleženje namreč upočasnjuje delovanje informacijskega sistema in povečuje čas dostopa do podatkov. Prav tako povečuje zahteve po količini prostora za shranjevanje. Količina revizijskih podatkov lahko raste z veliko hitrostjo, neustrezna implementacija beleženja v izvajalni proces pa lahko privede do nekontroliranega beleženja (npr. v primeru neskončne zanke). Podjetja morajo pri implementaciji beleženja implementirati ustrezne postopke kontrole, kjer je to potrebno, in zagotavljati zadostno količino sistemskih virov tudi iz vidika zagotavljanja učinkovitosti poizvedovanja po zbranih podatkih. Posebno pozornost je potrebno nameniti čiščenju podatkov, ki mora biti skladno z regulatornimi zahtevami (Audit trails, 2025).

5.4 *Skrbnništvo nad podatki*

Podatki morajo biti varno shranjeni, kar pomeni, da so šifrirani, dostop do njih pa je omejen. Do njih lahko dostopajo le pooblašчени uporabniki (Vicente, 2024).

Dnevnik obdelave mora vsebovati podatke o vrsti obdelave, času, datumu in jasni identifikaciji izvajalca in uporabnika, na katere se podatki nanašajo. Organizacije lahko vključijo tudi dodatne informacije ob upoštevanju ocene učinka. Ti podatki se uporabljajo izključno za preverjanje zakonitosti obdelave, izvajanje notranjega nadzora, odpravljanje napak in izpolnjevanje zakonskih zahtev. Na zahtevo pristojnega organa mora biti omogočen vpogled v te evidence (Zakon o varstvu osebnih podatkov /ZVOP-2/, 2022).

Dnevnik obdelav se praviloma hrani dve leti po koncu koledarskega leta, v katerem je bila obdelava izvedena oziroma kako določijo posamezni drugi zakoni (prav tam).

6 OBSTOJEČA IAM REŠITEV V OKOLJU ORACLE APEX

Obstoječa centralna rešitev IAM se uporablja za razvoj informacijskih rešitev. Namenjena je IT skrbnikom, razvojnikom ter lastnikom in skrbnikom procesov. Cilj IAM centralne rešitve je, da so dostopi in privilegiji v posameznih informacijskih rešitvah urejeni poenoteno in centralno (Cvar, 2022). Nahaja se v razvojnem okolju Oracle Apex. Imenuje se IAM in ureja aplikacije, module, uporabnike, profile, privilegije in vloge. V tem poglavju bomo natančneje predstavili obseg in omejitve, ki jih bomo upoštevali v tem diplomskem delu, opisali zgradbo in predstavili podatkovni model IAM centralne rešitve ter opisali poslovne procese (imenovano tudi poslovne dogodke), ki se izvajajo.

6.1 Obseg in omejitve

IAM lahko vsebuje različne funkcionalnosti: upravljanje programa, hramba uporabniških identitet, dodeljevanje uporabniških računov, upravljanje privilegijev in vlog, avtentikacijo in avtorizacijo, poročanje in evidenco sprememb ter operacij.

Obstoječa IAM rešitev večino teh funkcionalnosti že podpira. V diplomskem delu se bomo osredotočili na dopolnitev rešitve z beleženjem revizijskih sledi poslovnih dogodkov. Sledljivost sprememb je bila doslej implementirana na način vodenja zgodovinskih podatkov na posameznih baznih tabelah. Teh ne bomo spreminjali, ampak jih bomo primerjali z revizijskimi sledmi poslovnih dogodkov.

IAM rešitev prav tako ne ureja avtentikacije, saj to ureja malokodno razvojno okolje Oracle Apex. Oracle Apex prav tako ponuja možnost, da se ustvari lastna avtentikacijska shema, vendar bomo za potrebe te aplikacije ohranili originalno shemo *Application Express Accounts* (vsebuje preverjanje z uporabniškim imenom in geslom). Prav tako ima IAM rešitev že narejeno avtorizacijsko shemo.

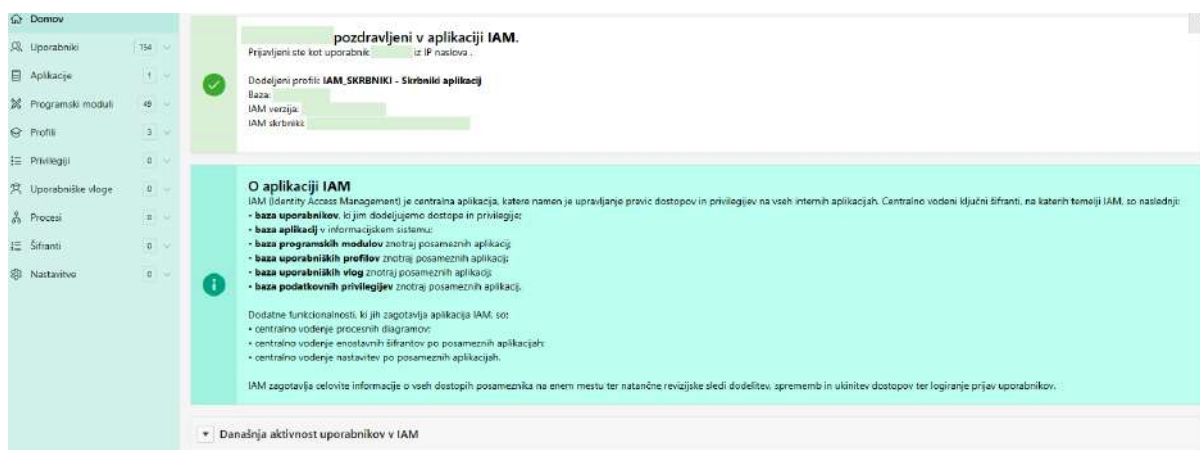
Pri obsegu in omejitvah je potrebno dodati, da se bomo osredotočili le na en del podatkovne baze. Pozornost bomo posvetili tabelam, ki so za potrebe diplomskega dela najpomembnejše. Prav tako bomo podatke v samih tabelah zakrili, anonimizirali ali uporabili psevdonimizirane podatke.

6.3 Zgradba aplikacije

Namen centralne rešitve IAM je upravljanje pravic dostopov in privilegijev na internih (Oracle) informacijskih rešitvah (Cvar, 2022). Kot je prikazano na sliki 22, IAM temelji na različnih centralno vodenih šifrantih in dodatnih funkcionalnosti, ki jih zagotavlja.

Prav tako lahko razberemo, da je IAM sestavljen iz več sklopov. Prikazana je domača stran (domov), ki prikazuje podatke od uporabnika, ki se nahaja v aplikaciji in podatke o aplikaciji.

Aplikacija vsebuje levi in desni stranski navigacijski meni. Meni na levi strani prikazuje glavne funkcionalnosti aplikacije: domov, uporabniki, aplikacije, programski moduli, profili, privilegiji, uporabniške vloge, procesi, šifranti in nastavitve. Desni navigacijski meni pa se pojavi samo na določenih straneh in vsebuje povezave do različnih pregledov in urejanj. Ti bodo predstavljeni v nadaljevanju.



Slika 22: Domača stran IAM rešitve

Vir: (Lastni vir)

6.3.1 Uporaba IAM centralne rešitve

IAM centralna rešitev predvideva uporabo in integracijo z lastnimi informacijskimi rešitvami, ni pa nanje omejena. Naloga skrbnikov aplikacij je dosledno upravljanje ključnih šifrantov za te informacijske rešitve. Med te spada določanje vseh uporabniških vlog in profilov, šifrant programskih modulov, prikaz podatkov, ki se bodo omejevali ... Prav tako je potrebno smiselno povezati uporabniške profile z moduli in privilegiji (prav tam).

Naloga razvojnikov IT rešitev pa je na osnovi definiranih avtorizacijskih pravil v lastne informacijske rešitve implementirati omejitve dostopov do programskih modulov ali podatkovnih poizvedb (Cvar, 2022).

6.3.2 Avtorizacijska shema obstoječe IAM rešitve

Uporabnik ima dostop do modulov, ki so povezani z njegovimi uporabniškimi profili. Uporablja se PL/SQL procedura za avtorizacijo dostopa, kot je prikazano na spodnjem posnetku zaslona (slika 23).

```
begin
  iam$sp_dostop(
    par_program      in  iam$programi.oznaka%type,
    par_uporabnik    in  iam$uporabnik_imena.uporabnisko_ime%type,
    out_smer         out varchar2,
    out_txt          out varchar2,
    out_sts          out pls_integer);
exception
  when others then
    out_sts := 1;
end;
```

Opis parametrov:

par_program	=> unikatna oznaka programa, npr. 'ZIV0_F_00086'
par_uporabnik	=> uporabniško ime, npr. 'CVARM'
out_smer	=> informacija, na kateri lokaciji se programski modul nahaja
out_txt	=> opis izhoda
out_sts	=> status izhoda; 0 – dostop dovoljen, različno od 0 – dostop ni dovoljen

Slika 23: Avtorizacijska shema - PL/SQL procedura

Vir: (Cvar, 2022)

Preko procedure se zabeleži vsak klic v IAM tabelo dostopov (beleženje dogodkov). V Oracle Apex je potrebno vklopiti avtorizacijsko shemo (prav tam).

Izgled avtorizacijske sheme (slika 24):

The screenshot displays the 'Authorization Scheme' configuration window. At the top, 'Scheme Type' is set to 'PL/SQL Function Returning Boolean'. The 'PL/SQL Function Body' section contains a PL/SQL function named 'do_dostop' that checks if a user is allowed to access a specific page. The function takes parameters for the program name, user, and page ID, and returns a boolean value based on a series of checks. Below the function body, there is a section for 'Identify error message displayed when scheme violated', which contains a message in Slovenian. The 'Evaluation Point' section has four radio buttons: 'Once per session', 'Once per page view' (which is selected), 'Once per component', and 'Always (No Caching)'. At the bottom, there is a 'Comments' section with a text area containing the comment 'Avtorizacijska shema IAM. Za vse strani (razen homepage)'.

```
declare
    b_tmp varchar2(1000);
    s_sts pls_integer;
    s_flag boolean;
begin
    begin
        do_dostop(
            par_program => '&APP_NAME.' || '-' || '&APP_PAGE_ID.',
            par_uporabnik => v('APP_USER'),
            par_vloga => null,
            out_smer => s_tmp,
            out_tx1 => s_tmp,
            out_sts => s_sts);
        exception
        when others then
            s_sts := 1;
        end;
        --
        if s_sts <> 0 then
            s_flag := false;
        else
            s_flag := true;
        end if;
        --
        return s_flag;
    end;
end;
```

Identify error message displayed when scheme violated

Dostop do strani &APP_NAME. - &APP_PAGE_ID. ni odobren.

<u>Podrobnejši opis:</u>

&P008_ERR_TXT.

Evaluation Point

Validate authorization scheme: ☐ Once per session ☒ Once per page view ☐ Once per component ☐ Always (No Caching)

Comments

Comments Avtorizacijska shema IAM. Za vse strani (razen homepage)

Slika 24: PL/SQL telo funkcije avtorizacijske sheme IAM rešitve

Vir: (Cvar, 2022)

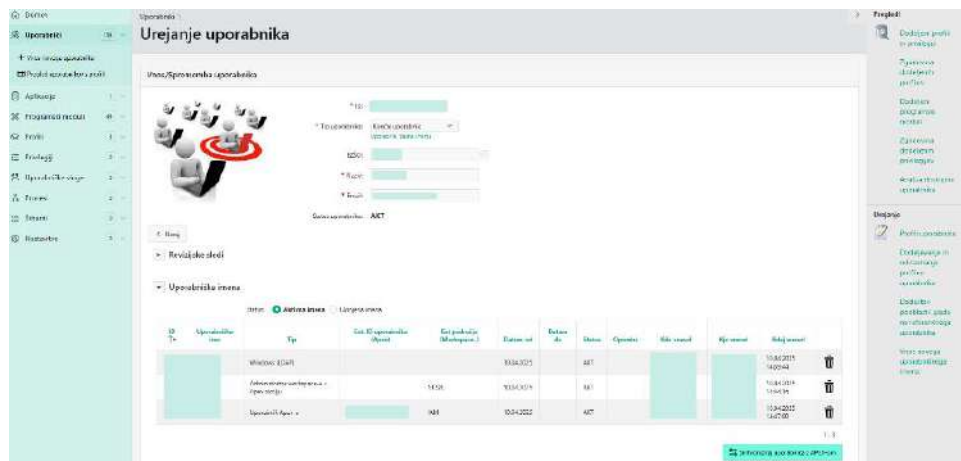
6.4 Uporabniški profili

Omejevanje dostopov uporabnikom temelji na dodeljevanju pravic uporabniškim profilom. Uporabnik ima lahko dodeljenih več profilov. Uporabnikom se pravice načeloma neposredno ne dodelijo. Znotraj IAM rešitve se uporabnikom dodeli smiselno število profilov. Prav tako se ne dodeljuje t. i. generičnih uporabnikov, ki bi jih uporabljalo več uslužbencev, saj se na tak način ne zagotavlja ustrezna revizijska sled (Cvar, 2022).

Dostop v IAM centralni rešitvi se omejuje na dostop do programskih modulov in dostop do podatkov (prav tam).

V IAM centralni rešitvi je možen pregled nad uporabniki. V sklopu *uporabniki* se prikaže pregled uporabnikov in njihovih profilov ter možnost vnosa novega uporabnika. Pri uporabnikih imamo v desnem navigacijskem meniju možnost pregleda in urejanja podatkov. Pri pregledu so možnosti: dodeljeni profili in privilegiji, zgodovina dodeljenih profilov,

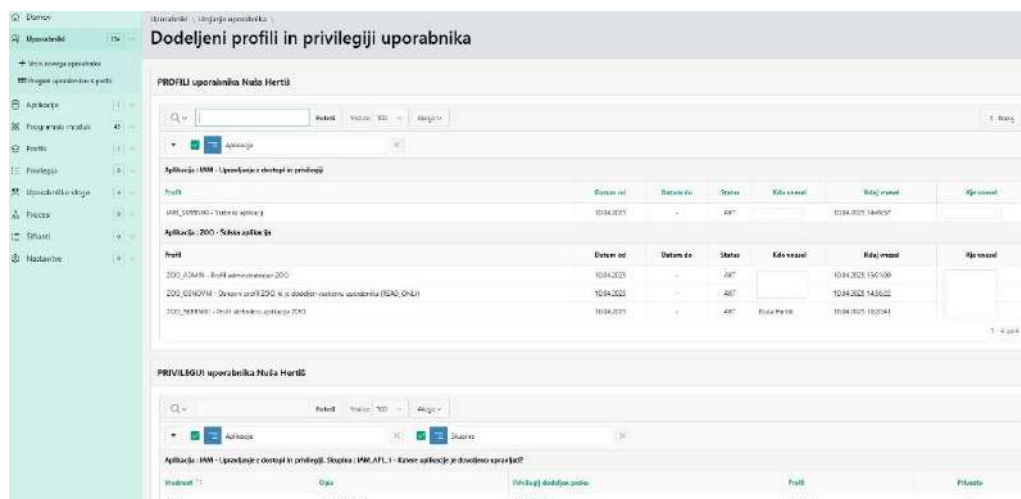
dodeljeni programski moduli, zgodovina dodeljenih privilegijev in analiza dostopov uporabnika. Pri urejanju pa: profili uporabnika, dodeljevanje in odvzemanje profilov uporabnika, dodelitev pooblastil glede na referenčnega uporabnika ter vnos novega uporabniškega imena.



Slika 25: Urejanje uporabnika

Vir: (Lastni vir)

Na sliki 26 je prikaz dodeljenih profilov in privilegijev uporabnika. Prikazano je torej, v katerih aplikacijah ima uporabnik dodeljen/e profil/e. Podoben izgled in lastnosti veljajo za ostale sklope.



Slika 26: Dodeljeni profili in privilegiji uporabnika

Vir: (Lastni vir)

6.5 Osnovni poslovni procesi

V nadaljevanju bomo našteali poslovne procese, t. i. poslovne dogodke, ki se izvajajo v IAM centralni rešitvi in kasneje natančneje opisali enega izmed njih.

Poslovni dogodki so:

- Registracija novega uporabnika,
- spreminjanje podatkov uporabnika,
- ukinitve uporabnika,
- registracija novega profila,
- spreminjanje podatkov profila,
- ukinitve profila,
- registracija nove vloge,
- spreminjanje podatkov vloge,
- ukinitve vloge,
- registracija novega privilegija,
- spreminjanje podatkov privilegija,
- ukinitve privilegija,
- dodelitev profila uporabniku,
- odvzem profila uporabniku,
- dodelitev uporabniške vloge profilu,
- odvzem uporabniške vloge profilu,
- dodelitev privilegija profilu,
- odvzem privilegija profilu,
- dodelitev dostopa do programskega modula profilu,
- odvzem dostopa do programskega modula profilu,
- dodelitev dostopa do aplikacije profilu,
- odvzem dostopa do aplikacije profilu,
- dodelitev dostopa do programskega modula vlogi,
- odvzem dostopa do programskega modula vlogi,
- dodelitev dostopa do aplikacije vlogi,
- odvzem dostopa do aplikacije vlogi.

7 NADGRADNJA IAM APEX APLIKACIJE

V diplomskem delu bomo nadgradili IAM rešitev v malokodnem razvojnem okolju Oracle Apex. IAM centralna rešitev je bila že podrobno predstavljena v prejšnjem poglavju, sedaj pa bomo predstavili še nadgradnjo. Implementirale se bodo revizijske sledi poslovnih dogodkov. Prav tako se bo v Oracle Apex implementirala generativna AI rešitev. Med seboj bomo primerjali dve AI storitvi: OpenAI in Cohere. Primerjali jih bomo po njuni učinkovitosti. Ob izvajanju nadgradnje in primerjave bomo preverjali postavljene hipoteze in jih potrdili oziroma ovrgli.

7.1 Identifikacija dogodkov za zapis revizijske sledi

Revizijske sledi bodo beležile poslovne dogodke, ki se izvajajo v IAM centralni rešitvi. Beležili se bodo samo določeni poslovni dogodki. Pri tem se bomo omejili na trinajst izbranih tabel v relacijskem modelu. Torej na tabele, v katere se vpisujejo podatki pri izvajanju izbranih poslovnih dogodkov. V diplomskem delu bomo predstavili izgled revizijske sledi na primeru poslovnega dogodka *dodelitev profila uporabniku*.

Ob dodelitvi profila uporabniku moramo upoštevati, da so podatki uporabnika in njegovega profila že v podatkovni bazi. Če podatki še niso podani, je potrebno najprej vnesti novega uporabnika. Podatek o uporabniku se vpiše v tabelo *IAM\$UPORABNIK*. Enako velja za profil. V tabelo *IAM\$PROFILI* se vnese nov profil. Ko sta uporabnik in profil vnesena, se naredi vnos v začasno tabelo in nato v tabelo *IAM\$UPORABNIK_PROFILI*.

Vsaka tabela ima svojo strukturno kopijo, zgodovinsko tabelo, v katero se hranijo zgodovinski zapisi. Prikazane so le tabele, potrebne za zapis podatkov v procesu *dodelitev profila uporabniku* (slika 27).

Kronološki pregled sprememb

Popravek	Spremenil	Kje	Velja od	Velja do	Aplikacija ID	Oznaka	Opis	Zap.št.	Navodila	Tip	Smer	Pošlji email	Datum od	Datum do	Status
12	Nuša Hertš		15.04.2025 08:38:51	-	12	ZOO-G-1	Gumb Create	1	Ta gumb pokliče Form, kjer lahko vstavivšcaron; nov podatek v tabelo	GUMB	1	-	11.04.2025	-	AKT
11	Nuša Hertš		15.04.2025 08:38:24	15.04.2025 08:38:51	12	ZOO-G-CREATE	Gumb Create	1	Ta gumb pokliče Form, kjer lahko vstavivšcaron; nov podatek v tabelo	GUMB	1	-	11.04.2025	-	AKT
10	Nuša Hertš		15.04.2025 08:37:02	15.04.2025 08:38:24	12	ZOO-G-CREATE	Gumb Create	1	Ta gumb pokliče Form, kjer lahko vstavivšcaron; nov podatek v tabelo	APEX	1	-	11.04.2025	-	AKT
9	Nuša Hertš		15.04.2025 08:30:54	15.04.2025 08:37:02	12	ZOO-G-1	Gumb Create	1	Ta gumb pokliče Form, kjer lahko vstavivšcaron; nov podatek v tabelo	APEX	1	-	11.04.2025	-	AKT
8	Nuša Hertš		15.04.2025 08:30:41	15.04.2025 08:30:54	12	ZOO-G-1	Gumb Create	10	Ta gumb pokliče Form, kjer lahko vstavivšcaron; nov podatek v tabelo	APEX	1	-	11.04.2025	-	AKT
7	Nuša Hertš		15.04.2025 08:23:38	15.04.2025 08:30:41	12	ZOO-G-1	Gumb Create	10	Ta gumb pokliče Form, kjer lahko vstavivšcaron; nov podatek v tabelo	GUMB	1	-	11.04.2025	-	AKT
6	Nuša Hertš		11.04.2025 15:29:09	15.04.2025 08:23:38	12	ZOO-001	Gumb Create	10	Ta gumb pokliče Form, kjer lahko vstavivšcaron; nov podatek v tabelo	GUMB	1	-	11.04.2025	-	AKT
5	Nuša Hertš		11.04.2025 15:27:18	11.04.2025 15:29:09	12	ZOO-CREATE	Gumb Create	10	Ta gumb pokliče Form, kjer lahko vstavivšcaron; nov podatek v tabelo	GUMB	1	-	11.04.2025	-	AKT

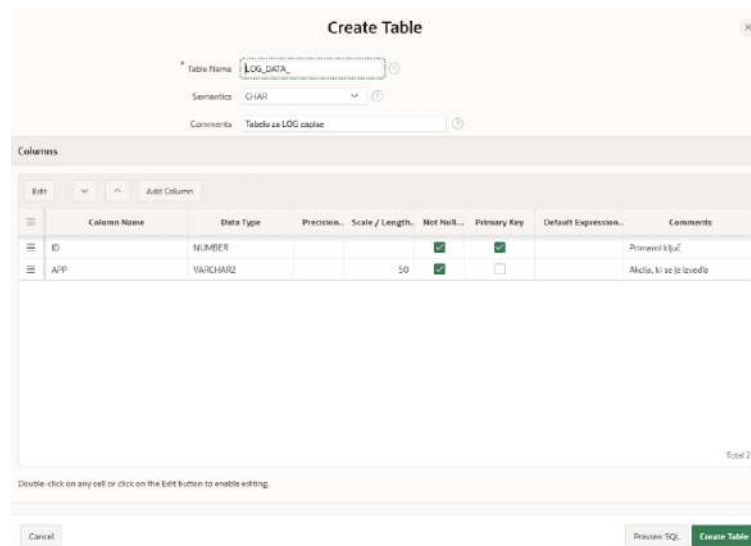
Slika 28: Zgodovinska tabela

Vir: (Lastni vir)

Pregled zgodovinskih tabel pa ni dovolj, da bi lahko ugotovili, kateri poslovni dogodek se je zgodil ali v katerem koraku poslovnega dogodka so nastale spremembe in kaj jih je sprožilo. Zato bomo v IAM rešitev implementirali beleženje dnevniških zapisov in revizijskih sledi.

7.2.1 Tabela dnevniških zapisov in revizijske sledi

Proces implementacije bomo prikazali v malokodnem razvojnem okolju Oracle Apex (2025). Najprej smo implementirali tabelo LOG_DATA, v katero se bodo shranjevali dnevniški zapisi. Tabele s pripadajočimi primarnimi in tujimi ključi ter "checks" je možno narediti v čarovniku (ni potrebe po uporabi SQL jezika). Čarovnik je prikazan na sliki 29.

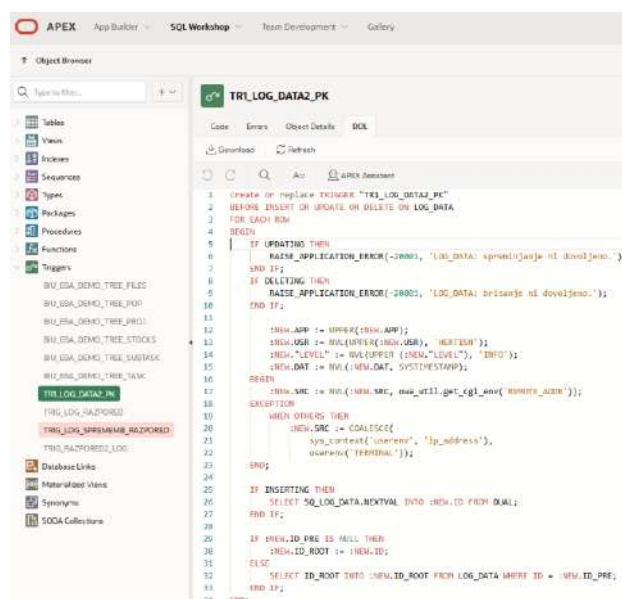


Slika 29: Čarovnik za ustvarjanje tabele

Vir: (Lastni vir)

Za zapis dnevniških zapisov in revizijskih sledi se ne sme uporabljati direkten vnos podatkov, spreminjanje podatkov ali brisanje le-teh. S tem bi se namreč izgubili konsistentnost podatkov. Potrebno je implementirati prožilce (angl. *triggers*) in pakete s procedurami in funkcijami, ki bodo poskrbeli za pravilen zapis podatkov. Teh s čarovnikom ni možno narediti. Potrebujemo znanje SQL in PL/SQL jezika. Na spodnjih posnetkih zaslona bodo prikazani DDL (angl. *data definition language*).

DDL prožilec (slika 30):

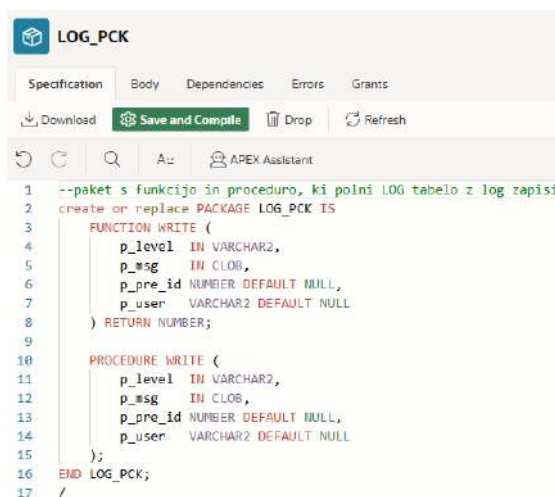


Slika 30: DDL prožilec

Vir: (Lastni vir)

Kot je razvidno s posnetkov zaslona (slika 31 in slika 32) sta ustvarjena dva paketa. Paket za zapis dnevniških zapisov (LOG_PCK) in paket za zapis revizijskih sledi (REV_PCK). Prikazana je specifikacija za oba paketa, ki smo jih naredili. Po specifikacijah smo implementirali še telo paketov.

DDL specifikacije paketa LOG_PCK (slika 31):



```

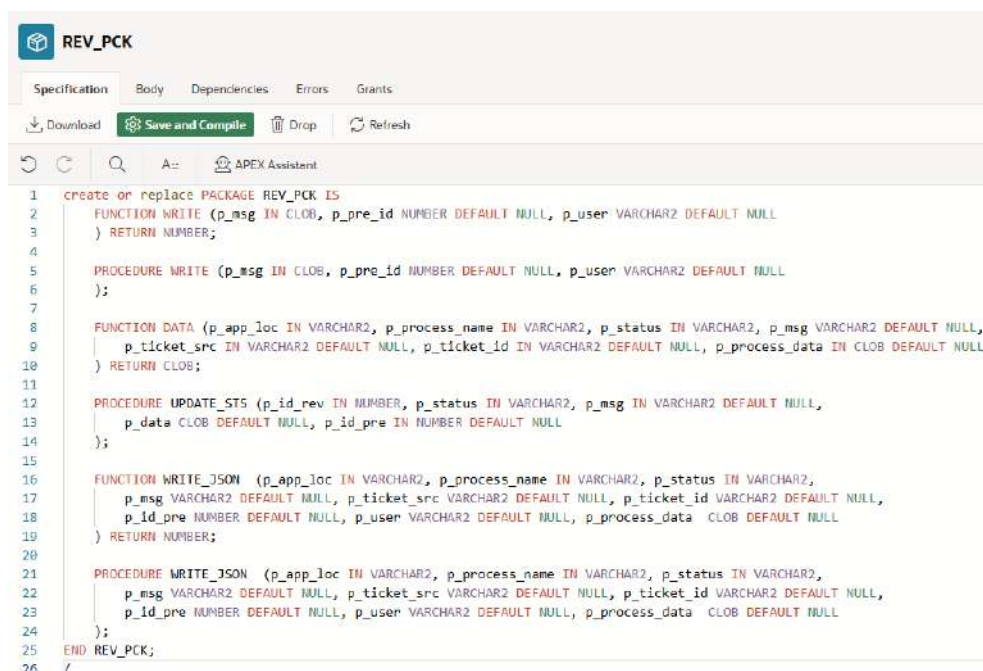
1  --paket s funkcijo in proceduro, ki polni LOG tabelo z log zapisi
2  create or replace PACKAGE LOG_PCK IS
3
4      FUNCTION WRITE (
5          p_level IN VARCHAR2,
6          p_msg IN CLOB,
7          p_pre_id NUMBER DEFAULT NULL,
8          p_user VARCHAR2 DEFAULT NULL
9      ) RETURN NUMBER;
10
11     PROCEDURE WRITE (
12         p_level IN VARCHAR2,
13         p_msg IN CLOB,
14         p_pre_id NUMBER DEFAULT NULL,
15         p_user VARCHAR2 DEFAULT NULL
16     );
17 END LOG_PCK;
18 /

```

Slika 31: DDL paket LOG_PCK

Vir: (Lastni vir)

DDL specifikacije paketa REV_PCK (slika 32):



```

1  create or replace PACKAGE REV_PCK IS
2
3      FUNCTION WRITE (p_msg IN CLOB, p_pre_id NUMBER DEFAULT NULL, p_user VARCHAR2 DEFAULT NULL)
4      RETURN NUMBER;
5
6      PROCEDURE WRITE (p_msg IN CLOB, p_pre_id NUMBER DEFAULT NULL, p_user VARCHAR2 DEFAULT NULL)
7      ;
8
9      FUNCTION DATA (p_app_loc IN VARCHAR2, p_process_name IN VARCHAR2, p_status IN VARCHAR2, p_msg VARCHAR2 DEFAULT NULL,
10         p_ticket_src IN VARCHAR2 DEFAULT NULL, p_ticket_id IN VARCHAR2 DEFAULT NULL, p_process_data IN CLOB DEFAULT NULL)
11      RETURN CLOB;
12
13     PROCEDURE UPDATE_STS (p_id_rev IN NUMBER, p_status IN VARCHAR2, p_msg IN VARCHAR2 DEFAULT NULL,
14         p_data CLOB DEFAULT NULL, p_id_pre IN NUMBER DEFAULT NULL)
15     ;
16
17     FUNCTION WRITE_JSON (p_app_loc IN VARCHAR2, p_process_name IN VARCHAR2, p_status IN VARCHAR2,
18         p_msg VARCHAR2 DEFAULT NULL, p_ticket_src VARCHAR2 DEFAULT NULL, p_ticket_id VARCHAR2 DEFAULT NULL,
19         p_id_pre NUMBER DEFAULT NULL, p_user VARCHAR2 DEFAULT NULL, p_process_data CLOB DEFAULT NULL)
20     RETURN NUMBER;
21
22     PROCEDURE WRITE_JSON (p_app_loc IN VARCHAR2, p_process_name IN VARCHAR2, p_status IN VARCHAR2,
23         p_msg VARCHAR2 DEFAULT NULL, p_ticket_src VARCHAR2 DEFAULT NULL, p_ticket_id VARCHAR2 DEFAULT NULL,
24         p_id_pre NUMBER DEFAULT NULL, p_user VARCHAR2 DEFAULT NULL, p_process_data CLOB DEFAULT NULL)
25     ;
26 END REV_PCK;
27 /

```

Slika 32: DDL paket REV_PCK

Vir: (Lastni vir)

Oba paketa ustvarjata zapise v tabelo *LOG_DATA*, vendar se med seboj razlikujeta:

- Dnevniški zapisi vključujejo akcijo, stopnjo pomembnosti (angl. *level*), sporočilo, predhodni ID (če obstaja) in podatke, ki se vpišejo s pomočjo prožilca (čas, oseba, kraj nastanka).
- Vsi zapisi, ki zapisujejo poslovne dogodke, imajo v stolpcu z oznako aplikacijskega vira (*APP*) vrednost, ki se dogovorjeno začne z "*REV.*".
- Zapis revizijske sledi vključujejo vse podatke, ki jih vsebuje dnevniški zapis in še dodatne podatke, ki so zapisani v JSON (angl. *JavaScript object notation*) obliki. Ti so:
 - lokacija aplikacije,
 - ime procesa,
 - status,
 - zahtevek za izvedbo,
 - dodatni podatki, ki so odvisni od specifikacije poslovnega dogodka.
- Revizijskim sledem dopišemo status, ki se spreminja.

Ker se dnevniški zapisi in revizijske sledi med seboj ločijo, smo ustvarili še dodatna dva pogleda (angl. *views*). En pogled (*REV_DATA*) prikazuje vse revizijske zapise, ki se v stolpcu *APP* začnejo z *REV.* Drug pogled pa prikazuje spremembe statusa revizijskih sledi (*REV_TIMELINE*). Torej prvi zapis sledi in zadnji zapis, ki je spremenil status in je prvemu zapisu podrejen.

Poslovne dogodke, ki jih bomo beležili, smo že določili. Prav tako imamo implementirano tabelo, v katero se bo beležilo prožilce in programske pakete, s pomočjo katerih se bo tabela polnila. Na primeru procedure za dodelitev/ukinitev profila uporabniku v IAM centralni rešitvi je prikazana implementacija revizijske sledi in dnevniškega zapisa. Koda je prikazana na sliki 33.

```

1 create or replace procedure azu_profil_uporabniki
2   (par_twp_id in number,
3    out_txt out varchar2,
4    out_sta out pls_integer) is
5   -- deklaracija spremenljivk
6   v_id number;
7   v1 EXCEPTION;
8   -- deklaracija kursorjev;
9   cursor profil_upo is
10    select id, id_org, uporabnik_id, profil_id, dat_do, dat_do,
11           zast, opomba, privzeto, zahteva_in
12    from iamuporabniki_profil_twp
13    where twp_id = par_twp_id;
14   begin
15     for n in profil_upo loop
16       -- n.id_org is not null
17       and n.zast = 'N' then
18       -- Ukinitve dostopa
19       -- najprej zapis revizije, nato se izvede update
20       v_id := REV_PCK.WRITE_JSON(
21         -- p_app      => 'IAM Apex',
22         -- p_level     => 'INFO',
23         p_app_loc   => 'IAM Apex',
24         p_process_name => 'Ukinitev profila uporabnika',
25         p_status     => 'Odnos',
26         p_msg        => 'Ukinitev profila uporabnika zaradi zahtevke',
27         p_ticket_src  => null,
28         p_ticket_id   => n.zahtev_id,
29         p_id_pre      => n.zahtev_id,
30         p_id_pre      => null,
31         p_user        => null,
32         p_process_data => TO_CLOB('{"uporabnik_id": ' || n.uporabnik_id || ', ' ||
33                                   '"profil_id": ' || n.profil_id || ', ' ||
34                                   '"opomba": ' || n.opomba || '}');
35       );
36       update iamuporabniki_profil
37       set dat_do = trunc(sysdate),
38           status = 'OKI'
39       where id = n.id_org;
40       -- dokonaj update_sta
41       REV_PCK.UPDATE_STS(
42         -- p_id_rev   => v_id,
43         -- p_app      => 'rev.pck.update_sta',
44         p_status     => 'COMPLETED',
45         p_msg        => 'Profil uspešno ukinjen uporabniku',
46         p_data       => TO_CLOB('{"opomba": ' || n.opomba || '}');
47       );
48     else
49       n.id_org is not null
50       and n.zast = 'D' then
51       -- Posodobitev datuma do
52       LOG_PCK.WRITE(
53         p_level      => 'INFO',
54         p_msg        => TO_CLOB('{"message": "Posodobljen datum do " || n.dat_do || '}'),
55         p_pre_id     => null,
56         p_user       => null
57       );
58       -- Posodobitev datuma do
59       update iamuporabniki_profil
60         set dat_do = n.dat_do
61         where id = n.id_org
62         and n1(dat_do, '19-DEC-2099') <> n1(n.dat_do, '19-DEC-2099');
63       elsif n.id_org is null
64       and n.zast = 'D'
65       and n.dat_do >= trunc(sysdate) then
66       -- Dodatek: Zapis revizijske sledi
67       v_id := REV_PCK.WRITE_JSON(
68         -- p_app_loc   => 'IAM Apex',
69         p_process_name => 'Dodajanje profila uporabniku',
70         p_status     => 'Odnos',
71         p_msg        => 'Dodajanje profila uporabnika zaradi zahtevke' || n.zahtev_id,
72         p_ticket_src  => null,
73         p_ticket_id   => n.zahtev_id,
74         p_id_pre      => null,
75         p_user        => null,
76         p_process_data => TO_CLOB('{"uporabnik_id": ' || n.uporabnik_id || ', ' ||
77                                   '"profil_id": ' || n.profil_id || ', ' ||
78                                   '"opomba": ' || n.opomba || '}');
79       );
80       insert into iamuporabniki_profil(uporabnik_id, profil_id, privzeto, opomba, dat_od, dat_do)
81       values (n.uporabnik_id, n.profil_id, n.privzeto, n.opomba, n.dat_od, n.dat_do);
82       -- Dodatek: Zapis revizijske sledi
83       REV_PCK.UPDATE_STS(
84         -- p_id_rev   => v_id,
85         p_status     => 'COMPLETED',
86         p_msg        => 'Profil uspešno dodan uporabniku',
87         p_data       => TO_CLOB('{"opomba": ' || n.opomba || '}');
88       );
89     exception
90     when others then
91       -- Dodatek: Zapis revizijske sledi
92       REV_PCK.UPDATE_STS(
93         p_id_rev     => v_id,
94         p_status     => 'Anaki',
95         p_msg        => 'Neuspešno izveden dogodek',
96         p_data       => TO_CLOB('{"opomba": ' || n.opomba || '}');
97       );
98       out_txt := 'Naznana napaka pri vnosu v tabelo iamuporabniki_profil. SQL: ' || sqlerrm;
99       raise #1;
100   end;
101   else
102     null;
103   end if;
104   and loop;
105   out_sta := 0;
106   out_txt := 'OK';
107   commit;
108   exception
109   when #1 then
110     out_sta := 1;
111   when others then
112     out_sta := 666;
113     out_txt := 'Naznana napaka - izvedba azu_profil_uporabniki - SQL: ' || sqlerrm;
114   end;

```

Slika 33: Koda procedure

Vir: (Lastni vir)

7.2.2 Programiranje v malokodnem razvojnem okolju Oracle Apex

V malokodnem razvojnem okolju Oracle Apex je možno oblikovati tabele v čarovniku (grafični vmesnik). S pomočjo čarovnika je prav tako možno narediti primarne ključe, tuje ključe, "check" in "unique constraint" (Tukaj je že potrebno vedeti osnove SQL, vendar si lahko pomagamo s podanim primerom.). Za pisanje procedur, funkcij, prožilcev in dodatnih funkcionalnosti pa je potrebno uporabiti SQL in PL/SQL jezik.

Torej se osnovne tabele s povezavami lahko naredi brez znanja osnovnega programiranja, vse nadaljnje nadgradnje pa ne. Za implementacijo smo torej morali poznati osnove programiranja.

7.3 Hranjenje

Revizijske sledi poslovnih dogodkov v IAM centralni rešitvi se bodo hranile na strežnikih podjetja. Podatki se bodo hranili 2 leti po nastanku, kot to določa predpis. Hranjenje podatkov in brisanje le-teh je implementirano s strani upraviteljev podatkov v podjetju.

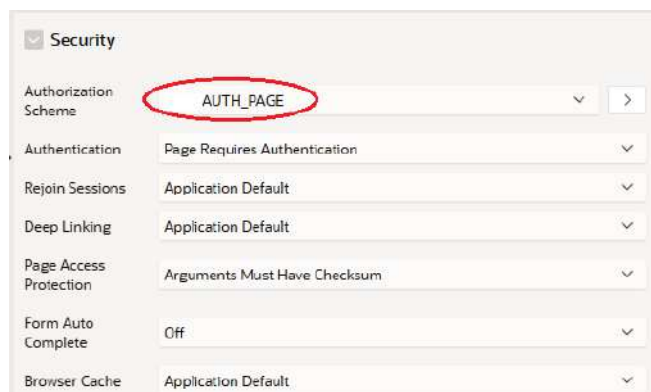
7.4 Dostop in upravljanje

V času nadgradnje IAM centralne rešitve bodo imeli dostop do revizijskih sledi le administratorji. V času implementacije revizijskih sledi je prav tako bilo možno upravljanje in spreminjanje tabele oziroma izbranih tabel v podatkovni bazi. Ko je bila implementacija izvedena, revizijskih sledi ni bilo več možno opravljati oziroma popravljati. To pomeni, da se sama tabela ne sme spreminjati, prav tako tudi ne podatkov v njej.

Vpogled v revizijske sledi imajo administratorji in revizorji. Vpogled bo prav tako po potrebi dodeljen uporabnikom. Uporabniki bodo imeli vpogled v revizijske sledi v aplikacijah, do katerih lahko dostopajo.

7.4.1 Avtorizacijska shema v IAM centralni rešitvi

V Oracle Apex IAM centralni rešitvi je implementirana avtorizacijska shema in omogočena Oracle Apex avtentikacijska shema. Kontroliran dostop do informacijskih rešitev je torej omogočen na tak način. Pri tem je poskrbljeno, da je implementirana avtorizacijska shema dodana na vseh straneh posamezne aplikacije. Na ta način lahko preverimo, ali ima določen uporabnik dostop do strani ali ne. Na sliki 34 je prikaz, kako se na posamezni strani doda avtorizacijska shema.



Slika 34: Urejevalnik strani - avtorizacijska shema

Vir: (Lastni vir)

Če ima uporabnik dodeljen dostop do strani, bo do njega lahko dostopal, sicer se mu bo prikazalo sporočilo, kot je prikazano na sliki 35.



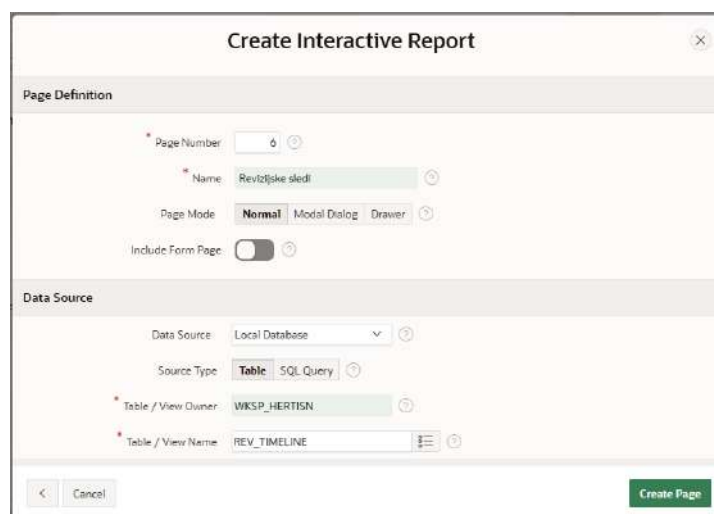
Slika 35: Dostop ni odobren

Vir: (Lastni vir)

Po natančnem pregledu smo ugotovili, da centralna rešitve IAM zagotavlja transparenten in kontroliran dostop. Z implementacijo revizijskih sledi smo torej to še izboljšali in omogočili različne oblike pregleda nad dogodki, ki se izvajajo v informacijskih rešitvah.

7.5 Pregledi

Za vpogled v revizijske sledi smo naredili novo stran v Oracle Apex. Stran vsebuje interaktivno poročilo revizijskih sledi. Pri implementaciji nove strani smo najprej s pomočjo čarovnika ustvarili novo stran s poročilom (slika 36).



Slika 36: Čarovnik ustvarjanje nove strani

Vir: (Lastni vir)

Izgled interaktivnega poročila, ki prikazuje revizijske sledi (pogled REV_TIMELINE), je prikazan na sliki 37. Pri tem je potrebno opozoriti, da so podatki psevdonimizirani.

Id	Id Last Step	Action	App Loc	Process	Status	Start At	Upd At	Ticket Id	Ticket Src	User Id	User Id	User Upd	User Upd	Upd Count
113	111	REVIZIJSKI_POKRITJE_JSON	RAM APX	REGISTRACIJA NOVEGA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
113	113	REVIZIJSKI_POKRITJE_JSON	RAM APX	REGISTRACIJA NOVEGA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
125	123	REVIZIJSKI_POKRITJE_JSON	RAM APX	REGISTRACIJA NOVEGA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
155		REVIZIJSKI_POKRITJE_JSON	RAM APX	REGISTRACIJA NOVEGA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
211	212	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
213	213	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
220	227	REVIZIJSKI_POKRITJE_JSON	RAM APX	REGISTRACIJA NOVEGA PROFILA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
234	235	REVIZIJSKI_POKRITJE_JSON	RAM APX	REGISTRACIJA NOVEGA PROFILA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
245	240	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
250	254	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	ERROR	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
317	318	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
325	326	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
333	334	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	ERROR	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
341	342	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
345	346	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
353	354	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	ERROR	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
361	362	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1
377	378	REVIZIJSKI_POKRITJE_JSON	RAM APX	ODLOŽITEV PROFILA UPORABNIKA	COMPLETED	30.06.2025 10:30	05.06.2025 10:30	TIC-3754M	E-Prod	NIH	91.232.239.254	H-TESTN	91.232.239.254	1

Slika 37: Prikaz revizijskih sledi

Vir: (Lastni vir)

Interaktivno poročilo je bilo možno ustvariti brez programiranja. Uporabili smo torej samo grafični vmesnik. S tem potrjujemo, da je osnovne komponente informacijskih rešitev možno narediti brez uporabe znanja programiranja. Tukaj je potrebno opozoriti, da je bila tema (torej oblika) same aplikacije predhodno definirana.

Kot je že bilo omenjeno, je centralna rešitev IAM že pred nadgradnjo implementirala zgodovinske tabele. Sedaj vsebuje še dnevniške zapise in revizijske sledi poslovnih dogodkov. S tem smo izboljšali transparentnost IAM. Revizorji in tudi sami razvojniki z novimi nadgradnjami vidijo različne spremembe, ki so se zgodile.

7.6 AI pomočnik

V Oracle Apex smo implementirali AI pomočnika, katerega namen je interpretirati revizijske sledi in podati odgovore na vprašanja, ki se na le-te nanašajo. Med seboj smo primerjali generativno AI rešitev OpenAI storitev in Cohere storitev.

7.6.1 Implementacija AI pomočnika

Implementacija sledi naslednjim fazam:

1. Ustvariti generativno AI storitev (Cohere in OpenAI)
2. Ustvariti AI konfiguracijo

3. Ustvariti RAG za AI konfiguracijo
4. Ustvariti modalni dialog AI pomočnika s pripadajočimi dinamičnimi akcijami in gumbi (Mohammed, 2025)

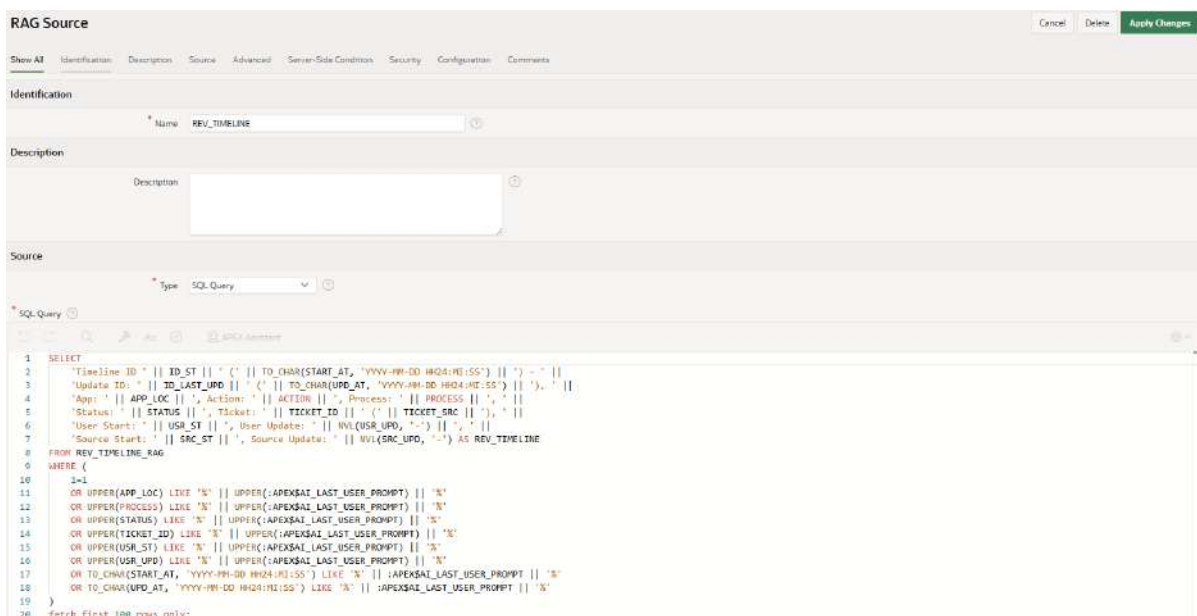
Faze od 2 do 4 je potrebno narediti dvakrat. Torej posebej za Cohere in za OpenAI. Na spodnjih posnetkih zaslona (slike 38–41) je prikaz faz na primeru storitve Cohere. API ključe za storitve smo pridobili z registracijo na spletnih straneh storitev.

Slika 38: Faza 1: Generativna AI storitev

Vir: (Lastni vir)

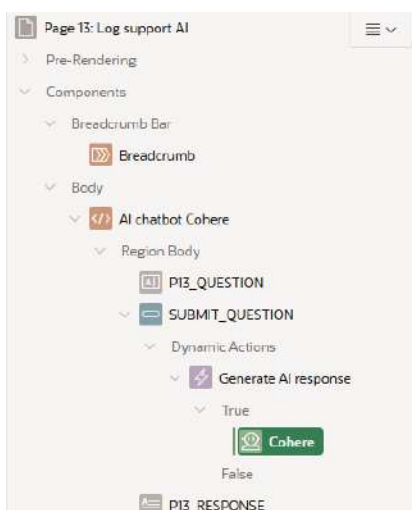
Slika 39: Faza 2: AI konfiguracija

Vir: (Lastni vir)



Slika 40: Faza 3: RAG za AI konfiguracijo

Vir: (Lastni vir)



Slika 41: Faza 4: AI pomočnik s pripadajočimi dinamičnimi akcijami in gumbi

Vir: (Lastni vir)

7.6.2 Kriteriji učinkovitosti

Za primerjalno vrednotenje učinkovitosti generativnih AI rešitev smo uporabili kriterije po mednarodnem standardu ISO/IEC 25010 (2025), ki določa značilnosti kakovosti programske opreme, in po kriterijih Jurafsky in Martin (2023). Vse kriterije smo prilagodili tako, da so relevantni za področje AI, specifično za vrednotenje in primerjavo storitev Cohere in OpenAI v malokodnem razvojnem okolju Oracle Apex na primeru vgrajenega pogovornega vmesnika.

V pogovornem vmesniku smo postavili vprašanja, ki se stopnjujejo po kompleksnosti. Vsa vprašanja so poizvedovalna vprašanja o podatkih revizijskih sledi.

Med značilnosti kakovosti ISO/IEC 25010 (2025) standarda spada učinkovitost delovanja (angl. *performance efficiency*), ki vsebuje tri podznačilnosti: časovno obnašanje, izraba virov in kapaciteta. Jurafsky in Martin (2023) prav tako podajata več kriterijev za ocenjevanje učinkovitosti: točnost, večjezičnost, uporabniška izkušnja, zmožnost razumevanja konteksta ... V diplomskem delu se bomo osredotočili na tri kriterije, za katere menimo, da so za naše potrebe najbolj primerni:

- Točnost oziroma kvaliteta: merilo, kako pravilno sistem odgovori na postavljeno vprašanje.
- Kapaciteta oziroma kompleksnost: zmožnost odgovora glede na dolžino in kompleksnost vprašanja.
- Izraba virov: poraba razdelilnikov za poizvedbe (angl. *tokens*, v nadaljevanju uporabljena angleška različica).

7.6.2.1 Točnost/kvaliteta

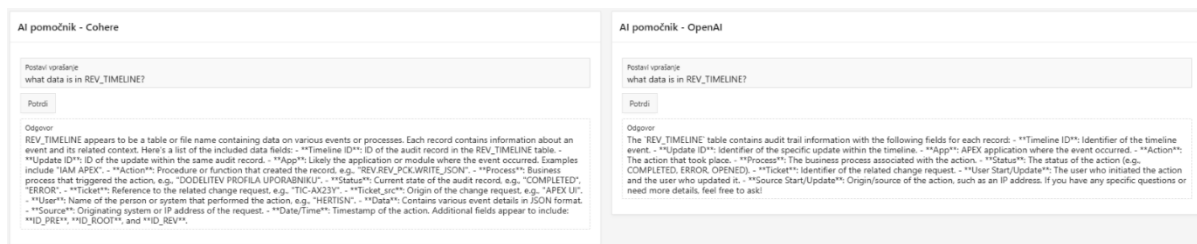
Točnost oziroma kvaliteta odgovorov se bo merila tako, da bomo ovrednotili pravilnost odgovorov na zastavljena vprašanja. Vprašali se bomo, ali sistem pravilno identificira in predstavi relevantne odgovore ter kako podrobno in uporabniku prijazni so odgovori. Odgovori bodo ocenjeni z oceno od 0 do 3. Tabela 1 prikazuje legendo numeričnih ocen in pripadajočih odgovorov, ki predstavljajo opisno oceno.

Tabela 1: Legenda ocen in odgovorov

Ocena	Odgovor
0	Nepravilen odgovor
1	Pomanjkljiv odgovor
2	Povprečen odgovor
3	Nadpovprečen odgovor

Vir: (Lastni vir)

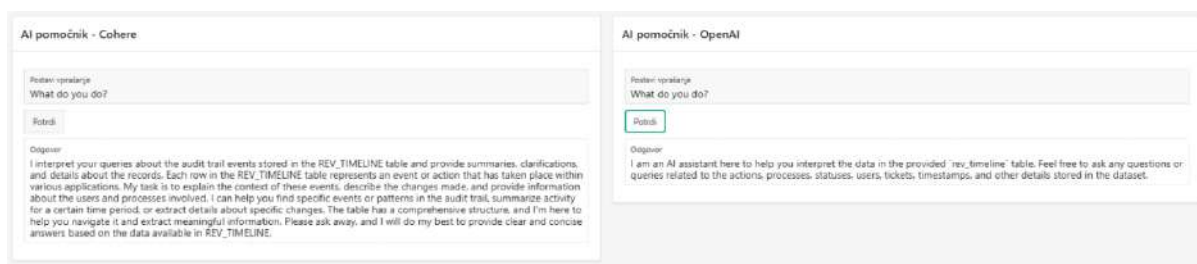
1. vprašanje: Na prvo vprašanje je po zastavljenih kriterijih boljši odgovor podal Cohere. Podani so namreč še primeri zapisov v tabeli. Cohere je podal *nadpovprečen odgovor* (3), OpenAI je podal *povprečen odgovor* (2).



Slika 42: AI pomočnik - 1. vprašanje

Vir: (Lastni vir)

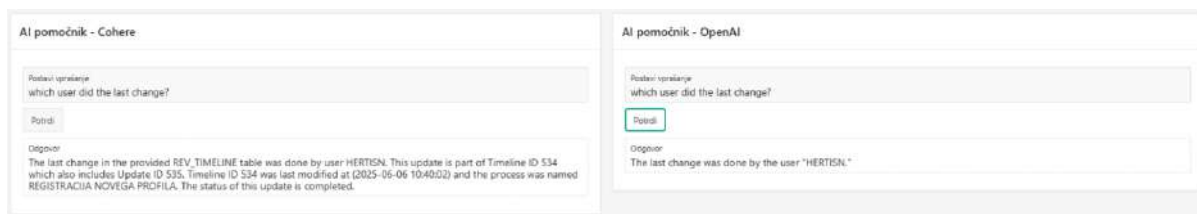
2. vprašanje: Na drugo vprašanje je bolje odgovoril Cohere. Iz podanega systemskega pozivnika (angl. *system prompt*) je razbral, kaj počne, in podal obsežna navodila, medtem ko je OpenAI podal krajši opis. Cohere je podal *nadpovprečen odgovor* (3), OpenAI je podal *povprečen odgovor* (2).



Slika 43: AI pomočnik - 2. vprašanje

Vir: (Lastni vir)

3. vprašanje: Na to vprašanje je ponovno bolj podrobno odgovoril Cohere. Podal je še podatke od zadnje spremembe, saj je predvideval, da želimo to izvedeti. Cohere je podal *nadpovprečen odgovor* (3), OpenAI je podal *povprečen odgovor* (2).



Slika 44: AI pomočnik - 3. vprašanje

Vir: (Lastni vir)

4. vprašanje: Tukaj sta oba odgovorila napačno. Ob času, ko smo postavili to vprašanje, je bilo v pogledu REV_TIMELINE 100 novih zapisov. Obe storitvi sta se zmedli in podali napačne podatke. Oba odgovora sta ocenjena z *nepravilen odgovor* (0).

AI pomočnik - Cohere	AI pomočnik - OpenAI
Postavi vprašanje How many changes were made today? Potrdi	Postavi vprašanje How many changes were made today? Potrdi
Odgovor There were 54 changes made today in the system. Each row in the timeline corresponds to one change, and each change can have multiple updates (indicated by the Update ID).	Odgovor There were a total of 68 changes made today in the audit trail stored in REV_TIMELINE.

Slika 45: AI pomočnik - 4. vprašanje

Vir: (Lastni vir)

5. vprašanje: Bolje je odgovoril Cohere. OpenAI namreč sploh ni podal odgovora. Cohere je podal *povprečen odgovor* (2), medtem ko dobi OpenAI oceno *nepravilen odgovor* (0).

AI pomočnik - Cohere	AI pomočnik - OpenAI
Postavi vprašanje Display all messages with status 'opened' in the last day. Potrdi	Postavi vprašanje Display all messages with status 'opened' in the last day. Potrdi
Odgovor Here are the messages with the status opened from the past day: **Timeline ID 470 (2025-06-06 10:38:20)** Process: UKINITEV UPORABNIKA Ticket: TIC-166FX (E-Pob) User Start: MC Source Start: 91.232.239.254 **Timeline ID 495 (2025-06-06 10:39:39)** Process: UKINITEV UPORABNIKA Ticket: TIC-QF7RK (E-Pob) User Start: TZ Source Start: 91.232.239.254 **Timeline ID 505 (2025-06-06 10:39:39)** Process: UKINITEV UPORABNIKA Ticket: TIC-908MU (E-Pob) User Start: MC Source Start: 91.232.239.254	Odgovor Based on the data provided, there are no messages in the last day with a status of 'opened'. The latest status for all entries is either 'ERROR' or 'COMPLETED'.

Slika 46: AI pomočnik - 5. vprašanje

Vir: (Lastni vir)

6. vprašanje: Ponovno je kvalitetnejši odgovor s strani storitve Cohere. Podana je še razlaga polj, ki so zapisana v slovenščini. Prav tako je navedel razlago za zapisom. OpenAI je navedel samo rezultate. Cohere je podal *nadpovprečen odgovor* (3), OpenAI je podal *povprečen odgovor* (2).

AI pomočnik - Cohere	AI pomočnik - OpenAI
Postavi vprašanje Fetch me data from table for id = 536 Potrdi	Postavi vprašanje Fetch me data from table for id = 536 Potrdi
Odgovor Here's a summary of the requested record and some context on the data. ## Record ID 536: **ID**:: 536 **ID_PRE**:: - **ID_ROOT**:: - **ID_REV**:: - **Action**:: REV.REV_PCK.WRITE/JSON **AppLoc**:: IAM APEX **Process**:: REGISTRACIJA NOVEGA PROFILA **Status**:: COMPLETED **Ticket_src**:: E-Pob **Ticket_id**:: TIC-7ARR4 **Level**:: Not recorded **Message**:: - **User HERTISN** **Date/Time**:: 2025-06-06 10:40:02 **Source**:: 91.232.239.254 **Data**:: { "app": "IAM APEX", "process": "REGISTRACIJA NOVEGA PROFILA", "status": "COMPLETED", "ticket_id": "TIC-7ARR4", "user_start": "NH", "user_update": "HERTISN", "source_start": "91.232.239.254", "source_update": "91.232.239.254", "id_rev": " " } ## Context: The record refers to an event in the IAM APEX application. The process, REGISTRACIJA NOVEGA PROFILA, translates to English as 'Registration of a new profile'. This particular audit trail records the successful registration of a new user profile, which is marked as completed. The origin of the request was the APEX user interface (UI), and the user who initiated and completed the process was 'HERTISN'.	Odgovor ```json [{ "Timeline ID": 536, "Update ID": 537, "App": "IAM APEX", "Action": "REV.REV_PCK.WRITE/JSON", "Process": "REGISTRACIJA NOVEGA PROFILA", "Status": "COMPLETED", "Ticket": "TIC-7ARR4 (E-Pob)", "User Start": "NH", "User Update": "HERTISN", "Source Start": "91.232.239.254", "Source Update": "91.232.239.254" }] ```

Slika 47: AI pomočnik - 6. vprašanje

Vir: (Lastni vir)

Kot je razvidno iz tabele 2, je storitev Cohere bolj točno in kvalitetnejše odgovorila na zastavljena vprašanja kot storitev OpenAI. Seštevek točk pri storitvi Cohere je 14, medtem ko je seštevek točk pri storitvi OpenAI enak 8.

Tabela 2: Kriterij točnost/kvaliteta

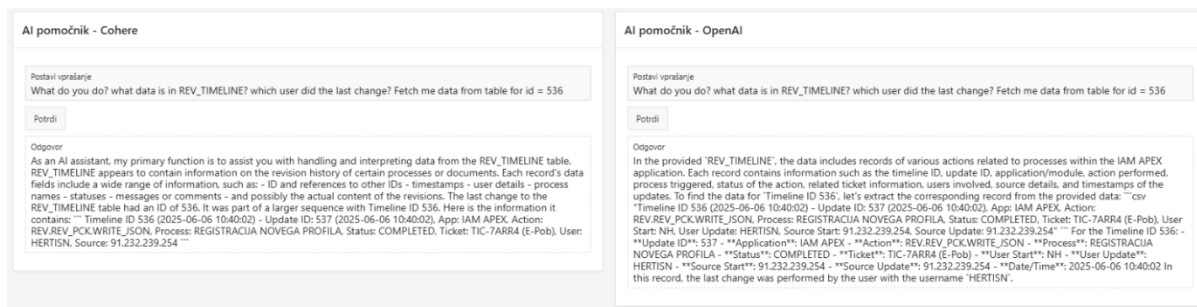
	OpenAI	Cohere
1. vprašanje	2	3
2. vprašanje	2	3
3. vprašanje	2	3
4. vprašanje	0	0
5. vprašanje	0	2
6. vprašanje	2	3
Skupaj	8	14

Vir: (Lastni vir)

7.6.2.2 Kapaciteta/kompleksnost

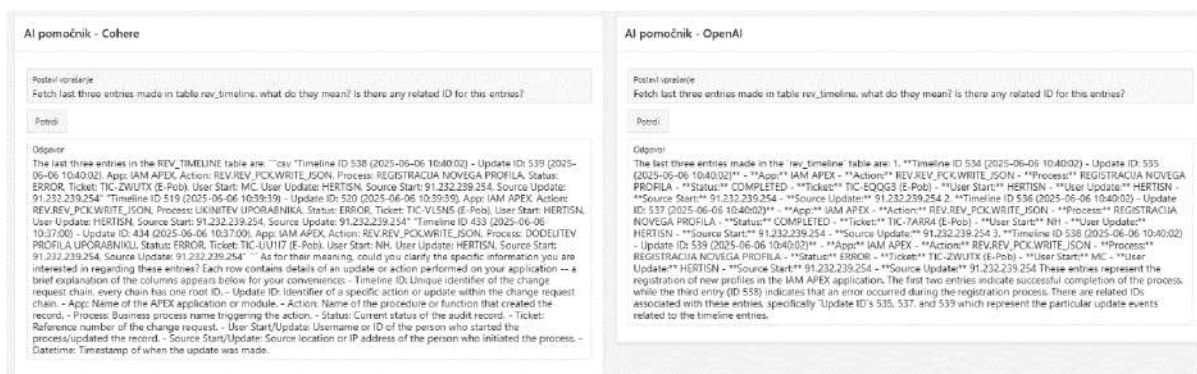
Kapaciteta in kompleksnost je merilo, ki meri zmožnost odgovora glede na dolžino in kompleksnost vprašanj. Preverili smo, kako odgovorita AI storitvi, če postavimo več vprašanj hkrati. Odgovori bodo ponovno ocenjeni z oceno od 0 do 3 (legenda je prikazana v tabeli 1).

Kot je razvidno s spodnjih dveh posnetkov zaslona (slika 48 in slika 49) so tukaj odgovori med seboj primerljivi. Obe storitvi sta natančno odgovorili na vsa postavljena vprašanja. Pri prvem posnetku zaslona so vprašanja enaka kot v prejšnjem podpoglavju. Pri drugem pa so postavljena tri nova vprašanja, ki se med seboj povezujejo. Pri obeh vprašanjih sta torej storitvi Cohere in OpenAI ocenjeni s *povprečen odgovor* (2).



Slika 48: AI pomočnik - 7. vprašanje

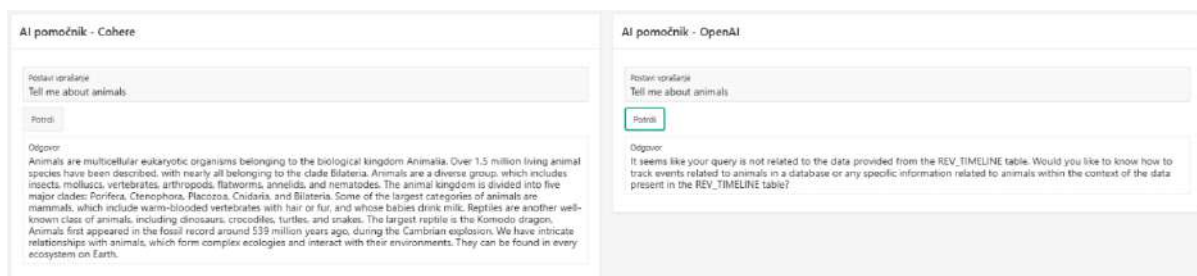
Vir: (Lastni vir)



Slika 49: AI pomočnik - 8. vprašanje

Vir: (Lastni vir)

Preverili smo tudi, ali storitvi upoštevata podan sistemski pozivnik (angl. *system prompt*), kjer smo navedli, naj se odgovarja samo na vprašanja, ki so povezana z revizijskimi sledmi. S tem smo preverili kompleksnost. Vprašali smo se, ali je storitev zmožna upoštevati več navodil ali jih ignorira. Kot je razvidno s slike 50, storitev Cohere odgovori na vprašanje in ne upošteva navodil, medtem ko OpenAI odgovori tako, kot so podana navodila v sistemskem pozivniku. Storitev Cohere prejme oceno *nepravilen odgovor* (0), storitev OpenAI pa je podal *povprečen odgovor* (2).



Slika 50: AI pomočnik - 9. vprašanje

Vir: (Lastni vir)

Storitev OpenAI je podala kvalitetnejše odgovore na kompleksnejša vprašanja kot storitev Cohere. Seštevek točk pri storitvi Cohere je 4, medtem ko je seštevek točk pri storitvi OpenAI enak 6 (tabela 3).

Tabela 3: Kriterij kapaciteta/kompleksnost

	OpenAI	Cohere
7. vprašanje	2	2
8. vprašanje	2	2
9. vprašanje	2	0
Skupaj	6	4

Vir: (Lastni vir)

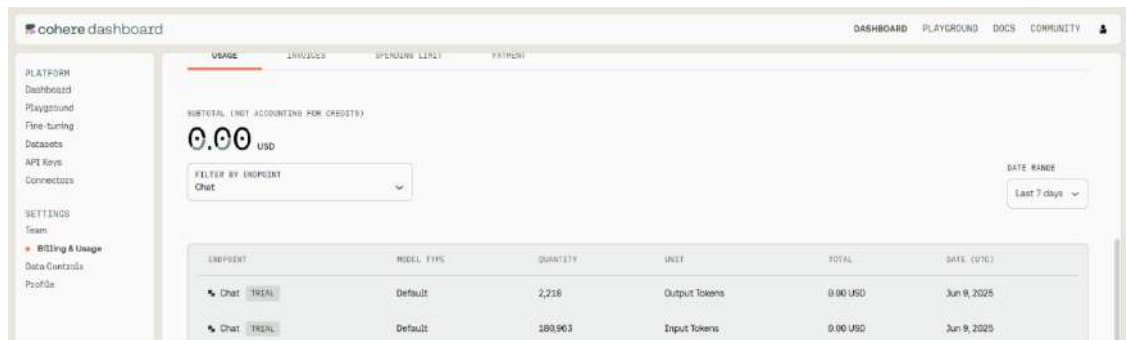
7.6.2.3 Izraba virov

Pri izrabi virov smo beležili, koliko tokenov se porabi za postavljena vprašanja. Tokeni, v kontekstu AI pomenijo besede, fraze oziroma znake. Vhodni tokeni so tokeni, ki jih AI prejme kot "vhod". To so torej vprašanja in podatki v sistemskem pozivniku. Izhodne tokene pa AI oblikuje v odgovore. Vsaka interakcija z AI šteje, koliko je porabljenih tokenov. To je osnova za obračunavanje cene uporabe AI, omejitev vprašanj in maksimalno število tokenov, ki se lahko porabijo naenkrat (OpenAI, 2025).

Porabo tokenov se meri na spletnih straneh Cohere in OpenAI, kjer se beležijo vhodni in izhodni tokeni. Primerjali bomo, koliko tokenov porabita storitvi za enaka vprašanja.

Kot je razvidno s spodnjih posnetkov zaslona (slika 51 in slika 52), Cohere porabi več vhodnih in izhodnih tokenov kot OpenAI. Torej je po učinkovitosti, glede na podkriterij izrabe virov, OpenAI bolj učinkovita storitev kot Cohere.

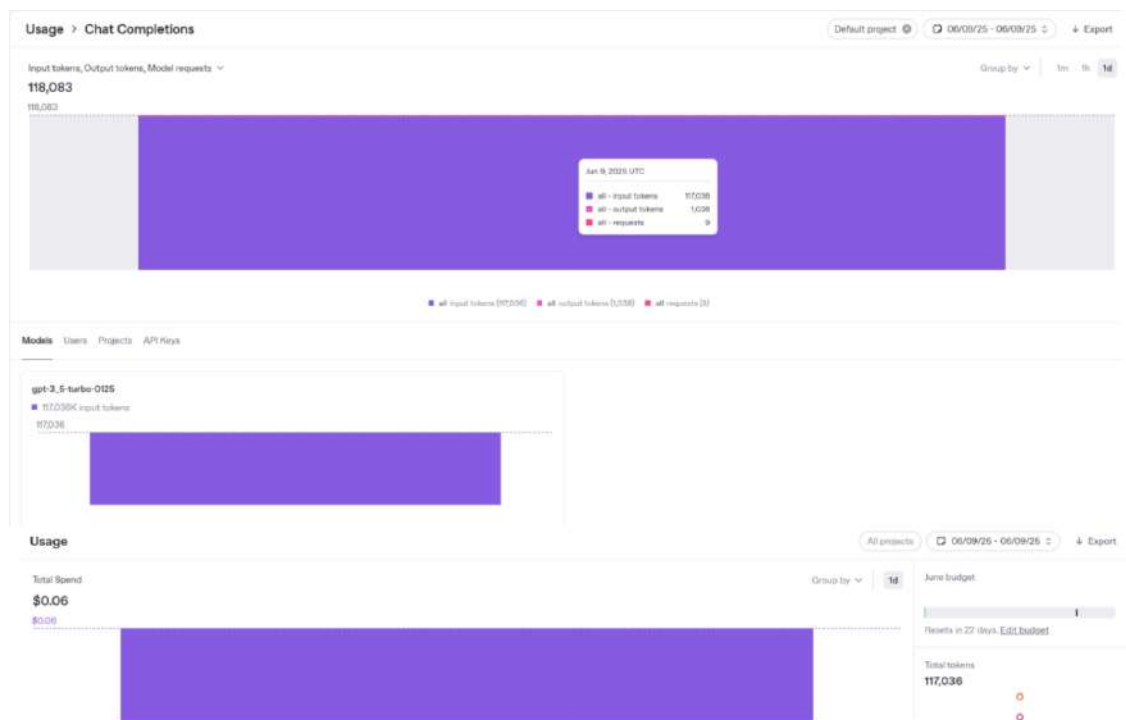
Cohere porabi:



Slika 51: Cohere - poraba tokenov

Vir: (Cohere, 2025)

OpenAI porabi:



Slika 52: OpenAI - poraba tokenov

Vir: (OpenAI, 2025)

Pričakovali smo, da bo število vhodnih tokenov približno enako, saj so bila postavljena vprašanja enaka, število izhodnih tokenov pa primerljivo z dolžino odgovorov posameznih storitev. Veliko razliko med količino tokenov predpisujemo temu, da Cohere in OpenAI uporabljata različne modele štetja tokenov. Obstajajo namreč različne oblike štetja tokenov, kot so BPE (angl. *byte-pair encoding*), metoda WordPiece in jezikovni model Unigram (Bostrom & Durrett, 2020).

7.6.3 Interpretacija

Kot je razvidno s tabele 4, je učinkovitejša storitev OpenAI. To sklepamo po tem, da je bila izbrana kot bolj primerna storitev v dveh od treh izbranih podkriterijev.

Tabela 4: Vrednotenje AI storitev

AI storitev/merila	OpenAI	Cohere
Točnost/kvaliteta		x
Kapaciteta/kompleksnost	x	
Izraba virov	x	

Vir: (Lastni vir)

Omejitve:

Pri interpretaciji rezultatov smo predpostavili, da imajo vsi podkriteriji enako pomembnost in jim zato pripisali enako težo. Prav tako je potrebno opozoriti, da modeli, ki smo jih uporabili, niso enaki. Poskusili smo uporabiti čim bolj primerljive modele. Pri storitvi Cohere smo uporabili njihov brezplačen AI model z imenom "*Command-r*". Te možnosti pri OpenAI ni, saj je potrebno pred uporabo modelov naložiti denar. V storitvi OpenAI smo izbrali AI model "*gpt-3.5-turbo*", ki je eden izmed najcenejših modelov in je po zmožnostih primerljiv modelu storitve Cohere. Kot je razvidno s posnetkov zaslona v poglavju 7.6.2.3 *Izraba virov*, smo z vprašanji porabili 0,06 \$, kar je 0,05 €. AI pomočnika smo prav tako spraševali v angleškem jeziku, saj nismo želeli, da jezik vpliva na rezultate (torej zmožnost modela, da prepozna in prevede jezike). Tukaj moramo prav tako podati, da je storitev Cohere bolj prilagodljiva pri podajanju virov RAG kot OpenAI. OpenAI namreč ne podpira možnosti, da je v viru RAG podan večji JSON in branje velike količine podatkov. Zato smo morali podati dodatne omejitve (omejitev pogleda in izpisa).

8 SKLEP

V diplomskem delu smo preučili sodobne informacijske rešitve v malokodnih razvojnih okoljih. Posebno pozornost smo posvetili platformi Oracle Apex, saj smo v tem okolju izvedli nadgradnjo IAM centralne rešitve. Glavni cilj diplomskega dela je bila izvedba nadgradnje, kar smo dosegli. Prav tako smo si postavili še dodatne cilje, ki so bili doseženi.

Ti cilji so bili, da podrobno predstavimo malokodno razvojno okolje Oracle Apex, kar smo naredili v začetnem teoretičnem izhodišču. Predstavili smo malokodna okolja, nato pa še predstavili specifične lastnosti okolja Oracle Apex. Naslednji cilj, ki smo si ga zastavili, je bil razložiti pojem IAM ter njegovo vlogo v informacijskih rešitvah. Tukaj smo najprej opisali, kaj je IAM in nato predstavili osnovne funkcionalne zahteve. Pri tem smo se še posebej osredotočili na avtorizacijsko in avtentikacijsko shemo, saj ti dve prispevata h kontroliranemu in transparentnemu dostopu do funkcionalnosti. Prav tako smo si zastavili cilj, da bomo predstavili pomen revizijskih sledi in jih implementirali v obstoječo IAM rešitev.

S pomočjo SQL in PL/SQL jezika smo nato ustvarili tabele in povezovalne funkcionalnosti ter implementirali revizijske sledi, ki beležijo poslovne dogodke. Ob tem smo prav tako raziskali, kako se AI uporablja v malokodnih razvojnih okoljih. V diplomskem delu smo se še posebej osredotočili, kako lahko AI pomaga uporabnikom, in ustvarili modalni dialog, kjer lahko uporabnik v naravnem jeziku poizveduje o revizijskih sledeh. Pri tem smo se osredotočali na primerjavo učinkovitosti različnih AI storitev.

Ob podrobni analizi že obstoječe IAM centralne rešitve in ob nadgradnji le-te smo potrdili ali ovrgli postavljene hipoteze.

Hipoteza 1: Izdelava poslovnih informacijskih rešitev v malokodnem razvojnem okolju Oracle Apex ne zahteva osnovnega znanja o programiranju.

Hipoteza 1 je delno potrjena. Ob nadgradnji IAM centralne rešitve smo namreč ugotovili, da se lahko osnovne strukture brez dodatnih komponent naredi popolnoma brez uporabe osnovnega znanja programiranja. Oracle Apex namreč ponuja čarovnike, s katerimi se gradijo tabele, primarni in tuji ključi in razne omejitve na poljih. Enako velja za ustvarjanje strani. Torej za ustvarjanje strani, ki vsebuje interaktivno poročilo, ni potrebe po znanju programiranja. Pri tem je potrebno opozoriti, da je ob gradnji informacijskih rešitev vseeno potrebno znanje o podatkovnih bazah, strukturah in povezavah med tabelami. S tem pa torej tudi osnovno znanje SQL jezika za razne poizvedbe, ki jih je potrebno vstaviti v čarovnik oziroma urejevalnik strani.

Torej hipoteze 1 ne moramo v celoti potrditi. Za potrebe nadgradnje, ki smo jo naredili v diplomskem delu, smo morali uporabiti znanje o SQL in PL/SQL jeziku. Napisali in implementirali smo več procedur, funkcij in prožilcev. Ugotovili smo, da je potrebno vsaj osnovno znanje programiranja, da bi ustvarili in/ali nadgradili informacijsko rešitev.

Hipoteza 2: Uporaba IAM rešitve zagotavlja transparenten in kontroliran dostop do funkcionalnosti v informacijskih rešitvah.

Hipoteza 2 je potrjena. Po podrobni analizi centralne rešitve IAM (upravljanje identitet in dostopov) ter pregledu in analizi dokumentacije o IAM rešitvi se hipoteza potrjuje. IAM centralna rešitev namreč vsebuje komponente, ki prispevajo h kontroliranemu in transparentnemu dostopu do funkcionalnosti v informacijskih rešitvah. To sta na primer avtentikacijska in avtorizacijska shema, ki omogočata kontroliran dostop. Transparentnost zagotavlja podroben prikaz podatkov v obliki nadzorne plošče, kjer je prikazano, kdo je do aplikacij dostopal. V zgodovinskih tabelah pa je zabeleženo, kakšne spremembe so se odvile nad podatki v podatkovni bazi. IAM je širok pojem in vsebuje še mnoge dodatne funkcionalnosti, ki jih v obstoječem okolju lahko dosežemo z nadgradnjami. Del nadgradnje smo izvedli v tem diplomskem delu. Dodali smo revizijske sledi, kar je še izboljšalo transparentnost IAM centralne rešitve.

Hipoteza 3: Hranjenje zgodovinskih podatkov v podatkovnih tabelah ne omogoča revizijskega vpogleda v izvajanje poslovnih procesov znotraj informacijske rešitve.

Hipoteza 3 je potrjena. Zgodovinske tabele hranijo tehnične spremembe, ki so se zgodile na podatkih v tabelah podatkovne baze. S tem ni zagotovljen celovit vpogled v poslovne dogodke, ki so na te spremembe vplivali, saj sledenje elementarnim spremembam podatkov ne more natančno rekonstruirati poslovnega dogodka. Še posebej, kadar se poslovni dogodki izvedejo v več korakih in vplivajo na spremembo podatkov na več tabelah, uporabljajočasne tabele Zgodovinske tabele ne navajajo podatka, zakaj se je posamezen zapis spremenil, kot to opisujejo podatki v revizijskih tabelah. S pomočjo revizijskih sledi je omogočeno, da revizor pridobi vse revizijske podatke iz ene same vrstice. Izve lahko, kateri poslovni dogodek se je izvedel, v kateri aplikaciji se je izvedel, kdo ga je sprožil in kako se je zaključil, kdaj se je zgodil, kakšen je bil namen in kakšen je bil končni status dogodka.

Hipoteza 4: Generativna AI rešitev storitve OpenAI je bolj učinkovita v primerjavi s storitvijo Cohere v malokodnem razvojnem okolju Oracle APEX na primeru vgrajenega pogovornega vmesnika.

Hipoteza 4 je potrjena. Izbrali smo tri podkriterije za učinkovitost, ki so bili točnost/kvaliteta, kapaciteta/kompleksnost in izraba virov. Vsi kriteriji so nosili enako težo. Pri dveh od treh podkriterijev je bila bolje ovrednotena storitev OpenAI. Ta dva kriterija sta kapaciteta/kompleksnost in izraba virov. Medtem ko je bila bolje ovrednotena storitev Cohere pri točnosti oziroma kvaliteti. Pri tem je potrebno opozoriti, da je bila storitev Cohere popolnoma brezplačna, medtem ko je storitev OpenAI zahtevala začetni vložek. Poizvedbe za namen tega diplomskega dela so stale 0,05 €.

Ključne ugotovitve diplomskega dela so, da malokodna razvojna okolja omogočajo hiter razvoj informacijskih rešitev in da za kompleksnejše funkcionalnosti zahtevajo znanje programskih jezikov. IAM predstavlja kontroliran in transparenten dostop do informacijskih rešitev, implementacija revizijske sledi pa transparentnost še bistveno izboljša. AI pomočniki lahko izboljšajo uporabniško izkušnjo, saj omogočajo poizvedbe v naravnem jeziku.

Diplomsko delo bi lahko prilagodili ali nadgradili tako, da bi uporabili še kakšne druge metode. Uporabili bi lahko kvalitativno metodo, pri čemer bi opravili intervju z razvijalcem, ki uporablja malokodna razvojna okolja. Če bi želeli pridobiti mnenje več oseb, bi lahko izvedli anketni vprašalnik. Vprašali bi programerje/razvijalce/uporabnike, kakšne so njihove izkušnje in mnenja o malokodnih razvojnih okoljih, AI, revizijskih sledih ... V tem diplomskem delu te metode nismo uporabili, saj smo dosegli vse zastavljene cilje z uporabo izbranih metod, torej uporaba dodatnih metod ne bi bila smiselna.

V diplomskem delu bi lahko raziskovano področje dodatno prilagodili oziroma razširili. Zapis revizijske sledi smo implementirali samo za izbrane poslovne dogodke. V realnem okolju bi to funkcionalnost bilo potrebno razširiti na vse poslovne dogodke. Prikazali bi lahko postopek vnosa podatkov na primeru nove aplikacije in kako uporabljati IAM centralno rešitev za omejevanje dostopa v tej aplikaciji. Prav tako bi lahko pri vrednotenju generativnih AI storitev uporabili dodatne kriterije za učinkovitost ali pa storitve vrednotili z drugimi kriteriji. Lahko bi primerjali več AI storitev, kot je OCI generativna AI storitev ali pa različne modele iste AI storitve. Pri tem ne smemo pozabiti, da smo uporabljali orodja podjetja Oracle Corporation. Raziskavo bi lahko razširili še na druga malokodna okolja in jih med seboj primerjali.

9 VIRI IN LITERATURA

- AI in Oracle APEX 24.1.* (2025). Pridobljeno iz Oracle Apex:
<https://apex.oracle.com/en/platform/features/whats-new-241/>
- AI in Oracle APEX 24.2.* (2025). Pridobljeno iz Oracle Apex :
<https://apex.oracle.com/en/platform/features/whats-new-242/>
- Alionco, A. (2024). *How will AI affect low-code/no-code development?* Pridobljeno iz Forbes:
<https://www.forbes.com/councils/forbestechcouncil/2024/09/25/how-will-ai-affect-low-codeno-code-development/>
- Apex oracle cloud.* (2025). Pridobljeno iz Oracle Apex :
<https://apex.oracle.com/en/platform/apex-oracle-cloud/>
- Architecture.* (2025). Pridobljeno iz Oracle Apex:
<https://apex.oracle.com/en/platform/architecture/>
- Audit trails.* (2025). Pridobljeno iz Data sunrise: <https://www.datasunrise.com/knowledge-center/audit-trails/>
- Bock, A., & Frank, U. (2021). Low-Code Platform. *Bus Inf Syst Eng* 63, 733–740. Pridobljeno iz <https://link.springer.com/article/10.1007/s12599-021-00726-8>
- Bostrom, K., & Durrett, G. (2020). Byte Pair Encoding is Suboptimal for Language Model. *Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics*. doi:10.18653/v1/2020.findings-emnlp.414
- Cimolini, P. (2017). *Oracle Application Express by Design: managing cost, schedule, and quality*. doi:10.1007/978-1-4842-2427-4.
- Cohere.* (2025). Pridobljeno iz <https://dashboard.cohere.com/>
- Cvar, M. (2022). *Standard Razvoja – IAM*. Maribor: Zavarovalnica Sava, d.d.
- Database concepts.* (2025). Pridobljeno iz Docs Oracle:
<https://docs.oracle.com/en/database/oracle/oracle-database/21/cncpt/introduction-to-oracle-database.html#GUID-A42A6EF0-20F8-4F4B-AFF7-09C100AE581E>
- Features.* (2025). Pridobljeno iz Oracle Apex:
<https://apex.oracle.com/en/platform/features/#security>

- Gillani, U. (2025). *What is Oracle? Uses of Oracle Database explained*. Pridobljeno iz Letsremotify: <https://letsremotify.com/blog/what-is-oracle-uses-of-oracle-database-explained/>
- Infografike. (2025). Pridobljeno iz Informacijski pooblaščenec: <https://www.ip-rs.si/publikacije/infografike>
- Introduction to ORDS. (2025). Pridobljeno iz Docs Oracle: <https://docs.oracle.com/en/database/oracle/oracle-rest-data-services/25.1/orddg/introduction-to-Oracle-REST-Data-Services.html#ORDDG-GUID-4D976DAE-8B24-41E3-81D5-B1C22E33287F>
- ISO/IEC 25010. (2025). Pridobljeno iz ISO 25000: <https://www.iso25000.com/index.php/en/iso-25000-standards/iso-25010>
- Jurafsky, D., & Martin, J. (2023). *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition (3rd ed. draft)*. Stanford university.
- Kauflig, J. (2021). *PaaS, IaaS, and SaaS: Making the Right Choice for Your Oracle Cloud Migration*. Pridobljeno iz Datavail: <https://www.datavail.com/blog/paas-iaas-and-saas-making-the-right-choice-for-your-oracle-cloud-migration/>
- Kosinski, M. (2024). *What is identity and access management (IAM)?* Pridobljeno iz IBM: <https://www.ibm.com/think/topics/identity-access-management>
- Leskovar, R., & Baggia, A. (Ured.). (2024). *Malokodno programiranje z APEX-om: Piročnik s praktičnimi primeri*. doi:<https://doi.org/10.18690/um.fov.6.2024>
- Martin, J. (2023). *Using low-code in your SDLC can accelerate delivery by 85%*. Pridobljeno iz HFS Research: <https://www.hfsresearch.com/research/low-code-agile-sdlc/>
- Medhi, A. (2025). *Iam vs pam*. Pridobljeno iz Securew2: <https://www.securew2.com/blog/iam-vs-pam>
- MiniOrange. (2025). *9 Essential Features for Identity and Access Management (IAM) solution*. Pridobljeno iz Miniorange: <https://www.miniorange.com/blog/iam-features/>
- Mohammed, T. (2025). *Building Retrieval Augmented Generation in Oracle APEX with AI Configurations and RAG Sources*. Pridobljeno iz Blogs Oracle: <https://blogs.oracle.com/apex/post/blog-ai-config-and-rag-sources>

- Mohindroo, A. (2024). *Announcing Oracle APEX 24.1 General Availability*. Pridobljeno iz Oracle Apex: <https://blogs.oracle.com/apex/post/announcing-oracle-apex-241>
- Mohorič, T. (1992). *Podatkovne baze*. Ljubljana: FER.
- Nilsson, N. J. (1980). *Principles of artificial intelligence*. Morgan Kaufmann.
- OpenAI. (2025). Pridobljeno iz <https://platform.openai.com/settings/organization/usage?startDate=2025-06-09>
- Oracle. (2025). *Oracle SQL Developer Data Modeler (verzija 24.3.) [programska oprema]*. Oracle Corporation.
- Oracle. (2025). *Oracle SQL Developer (verzija 24.3) [programska oprema]*. Oracle Corporation.
- Oracle APEX (verzija 24.2.) [spletna aplikacija]. (2025). Pridobljeno iz <https://apex.oracle.com/>
- Prinz, N., Rentrop, C., & Huber, M. (2021). *Low-Code Development Platforms – A Literature Review*. Pridobljeno iz AMCIS 2021 Proceedings: https://web.archive.org/web/20220801212143id_/https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1079&context=amcis2021
- Richardson, C., & Rymer, J. (2016). *Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms*. Pridobljeno iz Forrester: <https://www.vaard.com/media/1090/forrester-report-2016q2.pdf>
- Rožanec, A. (2017). *Baze podatkov*. Fakulteta za upravljanje, poslovanje in informatiko. Pridobljeno iz <http://www.vsnm.si/publikacije/arozanec/bp2017/index.html>
- Success stories. (2025). Pridobljeno iz Oracle Apex: <https://apex.oracle.com/en/solutions/success-stories/>
- Sweeney, P. (2024). *What is identity and access management? Guide to IAM*. Pridobljeno iz Techtarget: <https://www.techtarget.com/searchsecurity/definition/identity-access-management-IAM-system>
- Syed, A. (2024). Oracle APEX development best practices: Ensuring scalability and maintainability. *International Journal of Latest Research in Progress*, 5(4). doi:<https://doi.org/10.5281/zenodo.15026525>

- Vicente, V. (2024). *What Is an Audit Trail? Everything You Need to Know*. Pridobljeno iz Audit board: <https://auditboard.com/blog/what-is-an-audit-trail/>
- Vincekovič, L. (2024). *Revizija skladnosti varstva osebnih podatkov z GDPR in ZVOP-2 [magistrsko delo]*. Pridobljeno iz <https://dk.um.si/IzpisGradiva.php?id=87055>
- What is a database?* (2020). Pridobljeno iz Oracle: <https://www.oracle.com/database/what-is-database/>
- What is low-code.* (2025). Pridobljeno iz Microsoft: <https://www.microsoft.com/en-us/power-platform/products/power-apps/topics/low-code-no-code/what-is-low-code>
- Wong, B. (2024). *What is identity and access management (IAM)?* Pridobljeno iz Forbes: <https://www.forbes.com/advisor/business/software/identity-access-management/>
- Zakon o informacijski varnosti /ZInfV-1/. (2025). *Uradni list RS, št. 40/25*. Pridobljeno iz Uradni list Republike Slovenije: <https://pisrs.si/pregledPredpisa?id=ZAKO8934>
- Zakon o varstvu osebnih podatkov /ZVOP-2/. (2022). *Uradni list RS, št. 163/22*. Pridobljeno iz Uradni list Republike Slovenije: <https://pisrs.si/pregledPredpisa?id=ZAKO7959>
- Zakon o zavarovalništvu /ZZavar-1/. (2015). *Uradni list RS, št. 93/15*. Pridobljeno iz Uradni list Republike Slovenije: <https://pisrs.si/pregledPredpisa?id=ZAKO6183>
- Zeichick, A. (2024). *Low-code*. Pridobljeno iz Oracle: <https://www.oracle.com/application-development/low-code/>
- Zorko, T. (2025). *Interna predstavitev UCS*. Maribor: Zavarovalnica Sava, d. d.
- Zorman, S. (2024). *Interni zapiski predmeta Zbirka podatkov II*. Maribor: Academia Maribor.