1) VIŠJA STROKOVNA ŠOLA ACADEMIA

MARIBOR

RAZVOJ VARNE KOMUNIKACIJSKE PLATFORME NUTRICARE Z UPORABO OGRODIJ PHP: PRIMERJAVA OGRODIJ LARAVEL IN YII

Kandidatka: Veronika Tomovska

Vrsta študija: Izredni študijski program

Študijski program: Računalništvo

Mentor predavatelj: mag. Ervin Schaff

Mentor v podjetju: Blaž Glogovčan, dipl. inž. rač. in inf. tehnol. (UN)

Lektorica: Nuša Vešligaj, mag. prof. slov. in zgod.

Maribor, 2026

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Podpisana Veronika Tomovska sem avtorica diplomskega dela z naslovom Razvoj varne komunikacijske platforme NutriCare z uporabo ogrodij PHP: primerjava ogrodij Laravel in Yii, ki sem ga napisala pod mentorstvom Ervina Schaffa.

S svojim podpisom zagotavljam, da:

- je predloženo delo izključno rezultat mojega dela;
- sem poskrbela, 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, marec 2026 Podpis študenta:

ZAHVALA

Iskreno se zahvaljujem mentorju predavatelju Ervinu Schaffu za strokovno vodenje, potrpežljivost ter dragocene napotke in usmeritve pri izdelavi diplomskega dela. Njegovo znanje, izkušnje in konstruktivne pripombe so pomembno prispevale h kakovosti vsebine ter k uspešnemu zaključku študija.

Posebna zahvala je namenjena mentorju v podjetju Blažu Glogovčanu za strokovno pomoč, praktične nasvete in podporo pri izvedbi praktičnega dela. Njegova pripravljenost za sodelovanje in deljenje izkušenj mi je omogočila boljše razumevanje obravnavane problematike in uspešno povezovanje teoretičnega znanja s prakso.

Zahvaljujem se tudi lektorici Nuši Vešligaj za skrben jezikovni pregled diplomskega dela, strokovne pripombe ter prispevek k jezikovni in slogovni urejenosti besedila.

Iskreno se zahvaljujem vsem predavateljem Višje strokovne šole Academia, ki so mi v času študija posredovali dragoceno znanje, strokovne izkušnje in me spodbujali k nenehnemu strokovnemu in osebnemu razvoju.

Posebno zahvalo namenjam tudi svoji družini in prijateljem za razumevanje, potrpežljivost in moralno podporo v času študija in pisanja diplomskega dela. Njihova spodbuda in podpora sta mi bili v veliko pomoč pri premagovanju izzivov na poti do uspešnega zaključka študija.

POVZETEK

Spletne aplikacije so v današnjem digitalnem svetu bistvenega pomena za vsakodnevno komunikacijo, poslovanje in obdelavo občutljivih podatkov, zato je njihova varnost ena najpomembnejših razvojnih potreb. Ogrodja PHP (Hypertext Preprocessor), ki omogočajo hitrejši, bolj organiziran in varen razvoj kot ročna implementacija PHP, postajajo vse bolj priljubljena med razvijalci. Laravel in Yii, dve najbolj priljubljeni ogrodji, podpirata zasnovo MVC (Model-View-Controller) in imata vgrajene številne varnostne funkcije, kljub temu pa se razlikujeta po filozofiji razvoja, stopnji avtomatizacije in podpori skupnosti.

Cilj diplomskega dela je bil proučiti in primerjati ogrodji Laravel in Yii z vidika praktične uporabe, razvoja spletnih aplikacij in varnosti. Osnovi za raziskavo sta razvoj uporabne aplikacije NutriCare, ki olajša komunikacijo med uporabniki in prehranskimi strokovnjaki, ter pregled uradne dokumentacije, strokovne literature in varnostnih ukrepov. Aplikacija je bila ustvarjena z uporabo obeh ogrodij in je imela standardne funkcije, vključno z upravljanjem vlog, registracijo, komunikacijo in obrambo pred najpogostejšimi spletnimi grožnjami. Posebna pozornost je bila namenjena varnosti spletnih aplikacij, zlasti obrambi pred vrivanjem SQL (Structured Query Language), napadi XSS (Cross-Site Scripting) in CSRF (Cross-Site Request Forgery), upravljanju sej, šifriranju podatkov in preverjanju pristnosti uporabnikov. Ob pravilni uporabi ponujata obe ogrodji visoko stopnjo varnosti. Višja raven avtomatizacije in uporabniku prijazna metodologija naredita Laravel posebej priljubljen, saj zmanjšujeta tveganje za napake razvijalcev in omogočata hitrejši razvoj. Yii pa zagotavlja strožji, tehnični in zanesljivejši nadzor nad varnostnimi mehanizmi, zaradi česar je še posebej primeren za zapletene in dolgotrajne sisteme. Rezultati praktičnega dela so pokazali, da omogočajo Laravelov robusten ekosistem paketov, obsežna dokumentacija in živahna skupnost izjemno hiter razvoj, vendar se je Yii izkazal za zanesljivo, varno in stabilno ogrodje, ki daje razvijalcu večji nadzor nad arhitekturo aplikacije. Teoretična in praktična analiza nas vodi do zaključka, da je izbira ogrodja odvisna od zahtev projekta: Yii je bolj primeren za varnostno občutljive, kompleksne in dolgoročne rešitve, kot je aplikacija NutriCare, medtem ko je Laravel bolj primeren za sodobne spletne aplikacije in hitro izdelavo prototipov.

Ključne besede: *ogrodja PHP, Laravel, Yii, varnost spletnih aplikacij, arhitektura MVC*

ABSTRACT

Development of the secure NutriCare communication platform using PHP frameworks

Web applications play a crucial role in today's digital world for everyday communication, business operations, and the processing of sensitive data; therefore, their security represents one of the most important development requirements. PHP frameworks, which enable faster, more structured, and more secure development compared to manual PHP implementation, are becoming increasingly popular among developers. Although Laravel and Yii—two of the most widely used frameworks—both support the MVC architecture and include numerous built-in security features, they differ in their development philosophy, level of automation, and community support.The objective of this bachelor's thesis was to examine and compare the Laravel and Yii frameworks from the perspectives of practical use, web application development, and security. The development of the NutriCare application, which facilitates communication between users and nutrition experts, along with a review of official documentation, professional literature, and security mechanisms, served as the foundation of the research. The application was developed using both frameworks and included standard features such as role management, user registration and authentication, communication functionality, and protection against the most common web threats.Special attention was devoted to web application security, particularly protection against SQL injection, XSS and CSRF attacks, session management, data encryption, and user authentication. The study found that both frameworks provide a high level of security when used correctly. Laravel stands out due to its higher level of automation and user-friendly methodology, which reduce the risk of developer errors and enable faster development. Yii, on the other hand, offers stricter, more technical, and more reliable control over security mechanisms, making it especially suitable for complex and long-term systems.The results of the practical work demonstrated that Laravel's robust package ecosystem, extensive documentation, and active community enable exceptionally rapid development. However, Yii proved to be a reliable, secure, and stable framework that provides developers with greater control over application architecture. The theoretical and practical analysis leads to the conclusion that the choice of framework depends on project requirements: Yii is more suitable for security-sensitive, complex, and long-term solutions such as the NutriCare application, while Laravel is better suited for modern web applications and rapid prototyping.

Keywords: *PHP frameworks, Laravel, Yii, web application security, MVC architecture*

KAZALO VSEBINE

1 UVOD 10

1.1 Opis področja in opredelitev problema 10
1.2 Namen, cilji in osnovne trditve 10
1.3 Predpostavke in omejitve 12
1.4 Uporabljene raziskovalne metode 13

2 PHP OGRODJA IN SKUPNOSTI 15

2.1 Uvod v PHP ogrodja 15
2.2 Laravel 15
2.3 Yii 16

3 VARNOST SPLETNIH APLIKACIJ 19

3.1 Uvod v varnost spletnih aplikacij 19
3.2 Varnost v PHP 19
3.3 Varnost v ogrodju Laravel 20
3.4 Varnost v ogrodju Yii 20
3.5 Primerjava varnostnih mehanizmov 21
3.6 Vprašanja varstva podatkov in varnosti v spletnih zdravstvenih aplikacijah 22

4. RAZVOJ APLIKACIJE V PHP OGRODJIH 26

4.1 Razvoj v ogrodju Laravel 26
4.1.1 Priprava razvojnega okolja 26
4.1.2 Tehnologije v NutriCare 27
4.1.3 Struktura projekta 28
4.1.4 Implementirane funkcionalnosti 30
4.1.5 Prednosti in izzivi v Laravelu 32
4.2 Razvoj z ogrodjem Yii 37
4.2.1 Sistemska arhitektura in uporaba tehnologije 39
4.2.2 Yii MVC arhitektura 39
4.2.3 Izkušnje z razvojem Yii2 47

5 MOŽNOSTI ZA NADALJNIJ RAZVOJ APLIKACIJE NUTRICARE 52

6 SKLEP 55

VIRI IN LITERATURA 57

KAZALO SLIK

Slika 1: Število vprašanj, povezanih z ogrodjem Laravel 18
Slika 2: Razvoj aplikaciji NutriCare v ogrodju Laravel 26
Slika 3: Prikaz MVC arhitekture 29
Slika 4: Prikaz pogovora z aplikacijo NutriCare 31
Slika 5: Prikaz MVC arhitekture v ogrodju Yii 38
Slika 6: Diagram primerov uporabe aplikacije NutriCare 39
Slika 7: Prikaz aplikacije NutriCare v ogrodju Yii 44
Slika 8: Prikaz aplikacije NutriCare, narejene z ogrodjem Yii 46

KAZALO TABEL

Tabela 1:Primerjava PHP ogrodij Laravel in Yii 17
Tabela 2:Statistična primerjava virov in podpore za ogrodji Laravel in Yii 17
Tabela 3:Primerjava varnostnih mehanizmov med Laravel in Yii 21
Tabela 4:Tehnologije v NutriCare 27
Tabela 5:Primerjava razvoja aplikacij z ogrodjema Laravel in Yii 49

1 UVOD

1.1 Opis področja in opredelitev problema

Spletne aplikacije so bistven sestavni del vsakodnevne komunikacije, poslovanja in izmenjave informacij v današnjem digitalnem svetu. Varnost spletnih aplikacij postaja vse bolj pereče vprašanje zaradi naraščajočega števila uporabnikov in količine občutljivih podatkov, poslanih po internetu. Varnost spletnih aplikacij zahteva sistematičen pristop k razvoju programske opreme (Sommerville, Software Engineering.Pearson., 2016).

Dandanes razvijalci le redko ustvarjajo aplikacije iz nič, namesto tega uporabljajo ogrodja PHP, ki omogočajo hitrejši, zanesljivejši in bolj organiziran razvoj. Laravel in Yii sta dve izmed najbolj razširjenih ogrodij, ker ponujata organiziran način razvoja, podporo za MVC oblikovanje in integrirane varnostne funkcije.

Vsako ogrodje pa uporablja drugačno varnostno strategijo. Yii na primer ponuja vsestranski mehanizem filtriranja in validacije, medtem ko Laravel ponuja obrambo pred vrivanjem SQL, CSRF in XSS napadi. Osrednji temi raziskave sta, ali Laravel ponuja večjo varnost in enostavnejši razvoj kot Yii ter ročna implementacija PHP.

1.2 Namen, cilji in osnovne trditve

Namen raziskave

Namen diplomskega dela je **izvesti sistematično primerjavo** PHP ogrodij Laravel in Yii z vidika varnosti, enostavnosti razvoja ter praktične uporabnosti pri izgradnji sodobnih spletnih aplikacij. S praktično implementacijo aplikacije NutriCare v obeh ogrodjih želimo **preveriti**, katero ogrodje ponuja boljšo podporo razvijalcem, višjo raven vgrajene varnosti ter učinkovitejši razvojni proces.

Osrednji namen je tudi **prispevati k razumevanju**, kako izbira ogrodja vpliva na varnost končnega produkta, ter **ponuditi priporočila** razvijalcem pri odločanju med različnimi PHP ogrodji za projekte, kjer je varnost ključnega pomena (npr. zdravstvene aplikacije, finančni sistemi).

Cilji raziskave

C1: Analizirati in primerjati varnostne mehanizme ogrodij Laravel in Yii, s poudarkom na zaščiti pred napadi.

C2: Ovrednotiti kakovost dokumentacije, velikost in aktivnost razvijalskih skupnosti ter razpoložljivost učnih virov za obe ogrodji.

C3: Praktično razviti funkcionalno spletno aplikacijo (NutriCare) v obeh ogrodjih in dokumentirati razlike v razvojnem procesu.

C4: Primerjati enostavnost implementacije ključnih funkcionalnosti v obeh ogrodjih.

C5: Identificirati prednosti in slabosti vsakega ogrodja ter na podlagi ugotovitev pripraviti priporočila za izbiro ogrodja glede na tip projekta.

C6: Preveriti postavljene hipoteze z uporabo kombinacije kvalitativnih in kvantitativnih metod raziskovanja.

Osnovne trditve (hipoteze)

Na podlagi predhodnega pregleda literature in poznavanja obeh ogrodij postavljamo naslednje hipoteze:

H1: Laravel ima večjo in aktivnejšo skupnost razvijalcev v primerjavi z Yii, kar se odraža v večjem številu razširitev, boljši podpori in hitrejšem reševanju težav.

H2: Laravel ponuja boljšo in bolj dostopno dokumentacijo ter nižjo učno krivuljo za začetnike v primerjavi z Yii.

H3: Obe ogrodji, Laravel in Yii, zagotavljata primerljivo raven varnosti pri pravilni implementaciji, vendar Laravel ponuja več samodejnih varnostnih mehanizmov, ki zmanjšujejo možnost napak razvijalcev.

H4: Uporaba ogrodja Laravel poveča varnost spletne aplikacije v primerjavi z lastnoročno implementacijo v PHP-ju.

H5: Laravel omogoča hitrejši razvoj aplikacij zaradi večjega ekosistema vnaprej pripravljenih paketov in orodij v primerjavi z Yii.

1.3 Predpostavke in omejitve

Raziskava temelji na naslednjih **predpostavkah**:

1. Dostopnost dokumentacije: Predvidevamo, da so uradna dokumentacija in viri obeh ogrodij na voljo, posodobljeni in zaupanja vredni.
2. Stabilnost ogrodij: Predvidevamo, da med raziskavo ne bo prišlo do bistvenih sprememb arhitekture ali varnostnih ukrepov ogrodij, ki bi bistveno vplivale na ugotovitve.
3. Relevantnost za skupnost: Uporabljamo predpostavko, da informacije o velikosti in dejavnosti skupnosti, zbrane iz javno dostopnih virov, natančno prikazujejo trenutno stanje.
4. Primerljivost: Kljub temu da imata ogrodji različni razvojni filozofiji in ciljni skupini potrošnikov, menimo, da ju je mogoče objektivno primerjati.
5. Praktična uporabnost: Pričakujemo, da bo izdelava aplikacije NutriCare omogočila natančno oceno zmogljivosti izbranega ogrodja.

Omejitve raziskave so naslednje:

1. Časovne omejitve: Ker je obseg študije omejen na obdobje od julija 2025 do januarja 2026, ne bo obravnavala nobenega omembe vrednega razvoja, ki bi se lahko zgodil po tem datumu.
2. Obseg uporabe: Aplikacija NutriCare bo ustvarjena kot preprost prototip z omejeno funkcionalnostjo, zato ne bo vključevala vseh zapletenih funkcij, ki jih ima lahko popolnoma delujoča produkcijska aplikacija.
3. Osredotočenost na specifična ogrodja: Predmet te študije sta samo Laravel in Yii; druga ogrodja PHP, kot sta Symfony in CodeIgniter, niso primerjana.
4. Varnostne in razvojne omejitve: Čeprav dodatni dejavniki, kot sta zmogljivost ali skalabilnost v produkcijskem kontekstu, niso temeljito obravnavani, se raziskava osredotoča predvsem na varnostna vprašanja in težavnost razvoja.
5. Subjektivnost ocenjevanja: Glede na preteklo znanje in izkušnje raziskovalca so lahko nekateri dejavniki, *kot sta enostavnost uporabe* ali *kakovost dokumentacije*, precej subjektivni.
6. Jezikovna omejitev: Na voljo sta le literatura in dokumentacija v angleščini in slovenščini.

1.4 Uporabljene raziskovalne metode

Za dosego ciljev bomo uporabili kombinacijo kvalitativnih in kvantitativnih raziskovalnih tehnik:

1. Analiza literature

Metodično bomo pregledali knjige, znanstvene publikacije, uradne evidence, strokovno literaturo in internetne vire, ki obravnavajo:

- varnost ogrodja PHP in spletne aplikacije,
- primerjave ogrodij Yii in Laravel,
- najboljše prakse za ustvarjanje spletnih aplikacij,
- grožnje varnosti in obrambi (OWASP).

2. Primerjalna ocena

Laravel in Yii bomo primerjali organizirano z uporabo naslednjih standardov:

- aktivnost in velikost skupnosti,
- CSRF, XSS, vrivanje SQL in avtentikacija so primeri varnostnih ukrepov,
- krivulja učenja in kompleksnost razvoja,
- podpora in dokumentacija,
- ekosistem paketov in razširitev,
- prilagodljivost in zmogljivost.

Metodologija: Pregled forumov in razvojnih skupnosti, analiza statistike GitHub in uporaba primerjalnih tabel.

3. Študija primera

Za oceno praktične uporabnosti izbranega ogrodja bomo ustvarili uporabno aplikacijo NutriCare, ki bo služila kot študija primera. Aplikacija bo sestavljena iz:

- sistema za registracijo in preverjanje pristnosti uporabnikov,
- vmesnika za komunikacijo med uporabniki in svetovalci,
- preprostega upravljanja profilov in podatkov,

- uvedbe varnostnih ukrepov.

Metodologije: Iterativno testiranje funkcionalnosti, agilen pristop k razvoju ter dokumentiranje težav in prednosti, ki so se pojavile med procesom.

4. Eksperimentalni pristop

V praktičnem delu bomo ocenili varnostne funkcije obeh ogrodij, da bi potrdili njuno učinkovitost:

- simulirane bodo osnovne varnostne grožnje,
- sledili bodo preverjanje odziva ogrodja na netočne vhodne podatke,
- pregled varnostnih parametrov in ustvarjene kode.

5. Analiza kvalitativnih podatkov

Izvedli bomo kvalitativno oceno uporabniške izkušnje, jasnosti dokumentacije, dostopnosti učnih virov in odzivnosti skupnosti na podlagi izkušenj z razvojem aplikacij in pregledom dokumentacije v resničnem svetu.

6. Interpretacija in sinteza rezultatov

Na koncu diplomskega dela bomo združili in analizirali vse zbrane informacije. Nato bomo potrdili teorijo in ponudili predloge za izbiro najboljšega ogrodja glede na vrsto projekta (Sommerville, Software engineering, 2016).

2 PHP OGRODJA IN SKUPNOSTI

2.1 Uvod v ogrodja PHP

Razvoj spletnih aplikacij je v zadnjih dvajsetih letih zrasel in postal veliko bolj standardiziran. Zaradi enostavne uporabe, združljivosti s strežniki in obsežne skupnosti razvijalcev je PHP (Hypertext Preprocessor) še vedno eden najbolj priljubljenih programskih jezikov v industriji spletnega razvoja.

Sčasoma je bilo razvitih veliko ogrodij, ki omogočajo ponovno uporabo kode, izboljšano strukturo projektov in integrirane varnostne metode, da bi razvoj postal enostavnejši in bolj strukturiran. Med njimi sta še posebej pomembna Laravel in Yii, saj sta si ustvarila ime kot najboljši možnosti za ustvarjanje varnih, najsodobnejših in učinkovitih spletnih aplikacij.

Pri izbiri uveljavljene tehnologije za določen projekt je ključnega pomena razumevanje skupnosti posameznega ogrodja, saj ima vsako svojo filozofijo, arhitekturo in uporabniško bazo (Fowler M. , 2003).

2.2 Laravel

Taylor Otwell je leta 2011 razvil odprtokodno ogrodje Laravel, da bi odpravil pomanjkljivosti že obstoječih ogrodij PHP, kot je CodeIgniter. Arhitektura MVC (Model–View–Controller), na kateri je zgrajen Laravel, omogoča ločeno delitev upravljanja podatkov, logike in uporabniškega vmesnika (Otwell, 2024).

Med glavnimi funkcijami Laravela so:

- Eloquent ORM, dovršen sistem za upravljanje baz podatkov;
- Artisan CLI, orodje ukazne vrstice za avtomatizacijo opravil;
- mehanizem za predloge Blade, intuitiven sistem za razvoj predlog;
- vmesna programska oprema, plast za preverjanje in filtriranje zahtev, ki izboljšuje varnost;
- integracija s tehnologijami, ki omogočajo učinkovito vzdrževanje in skalabilnost aplikacij, kot so Laravel Forge, Nova, Horizon in Vapor.

Ena največjih skupnosti razvijalcev PHP na svetu trenutno obstaja na Laravelu. Uporablja ga več kot deset milijonov spletnih mest, skupnost pa sestavljajo razvijalci, podjetja, mentorji in izobraževalne platforme, kaže raziskava Laravel Developer Survey 2025 (Adevait, 2025).

Laravel skupnost ima več kot 10 milijonov uporabnikov (Community, 2025), kar potrjuje tudi raziskava na Stack Overflow (Stack Overflow, 2026), kjer je Laravel trenutno vodilno PHP ogrodje. Poleg tega razvijalci sodelujejo pri razvoju osrednjega ogrodja in si izmenjujejo izkušnje v aktivnih kanalih na GitHubu, Discordu in Redditu.

Prednosti skupnosti Laravel vključujejo:

- visoko raven angažiranosti in odzivnosti uporabnikov,
- bogastvo učnih virov (Laracasts, Laravel News),
- redne nadgradnje in stalno podporo,
- močan poudarek na varnosti in uporabnosti (Laravel, 2025).

2.3 Yii

Ogrodje Yii, kar pomeni Yes It Is! (Da, to je!), je še eno odprtokodno ogrodje PHP s poudarkom na varnosti, skalabilnosti in visoki zmogljivosti. Od svoje prve izdaje leta 2008 je ogrodje postalo ena najbolj zanesljivih možnosti za ustvarjanje zapletenih spletnih sistemov.

Z uporabo paradigme MVC ponuja Yii široko paleto vgrajenih zmogljivosti, vključno s sistemom za optimizacijo predpomnilnika in hitrosti, varnostnimi filtri za preprečevanje napadov SQL, XSS in CSRF, avtomatiziranim generiranjem kode (orodje Gii) in modularnim ogrodjem za preproste razširitve in prilagoditve.

Kljub svoji majhnosti je skupnost Yii zelo profesionalna. Yii Framework ima približno 540.000 registriranih uporabnikov (Yii Framework, 2024), medtem ko SimilarWeb (2025) ocenjuje, da ogrodje uporablja približno 550.000 spletnih mest po vsem svetu.

Forumi, repozitoriji GitHub in večjezična dokumentacija predstavljajo dejavnosti skupnosti. Razvijalci lahko prispevajo k izboljšavam jedra in razširitev na pogostih dogodkih, ki jih gosti fundacija Yii.

Prednosti skupnosti Yii vključujejo:

- stabilno in visokokakovostno dokumentacijo,
- tehnično usposobljene uporabnike,
- poudarek na varnosti in zmogljivosti,
- stabilen razvoj in združljivost različic.

Tabela 1: Primerjava PHP ogrodij Laravel in Yii

Kriteriji	Laravel	Yii
Velikost skupnosti	Zelo velika	Manjša (okoli 540.000 članov), vendar strokovna
Aktivnost	Visoka (pogoste konference, posodobitve, video vodiči)	Zmerna (forumi in GitHub razprave)
Dokumentacija	Obsežna (veliko primerov in vodiči)	Zelo jasna, tehnično natančna
Razvojni cikel	Pogoste nove različice, hitre inovacije	Počasnejši, stabilen razvoj
Ekosistem ogrodij	Zelo bogat	Omejen
Podpora	Uradni forumi, Laracasts, Discord	Uradni forumi in GitHub skupnost

Vir: (Framework, 2024), (Laravel, 2025), (Overflow, 2026)

Iz analize lahko sklepamo, da Yii ponuja boljšo stabilnost, varnost in tehnično doslednost, medtem ko Laravel ponuja večjo dostopnost, hitrejši razvoj in širšo podporo. Odločitev med obema ogrodjema temelji na zahtevah projekta; Laravel je primernejši, če je cilj hitra uvedba in pomoč skupnosti, Yii pa je boljša možnost, če sta cilja varnost in zanesljivost pri dolgoročnih projektih.

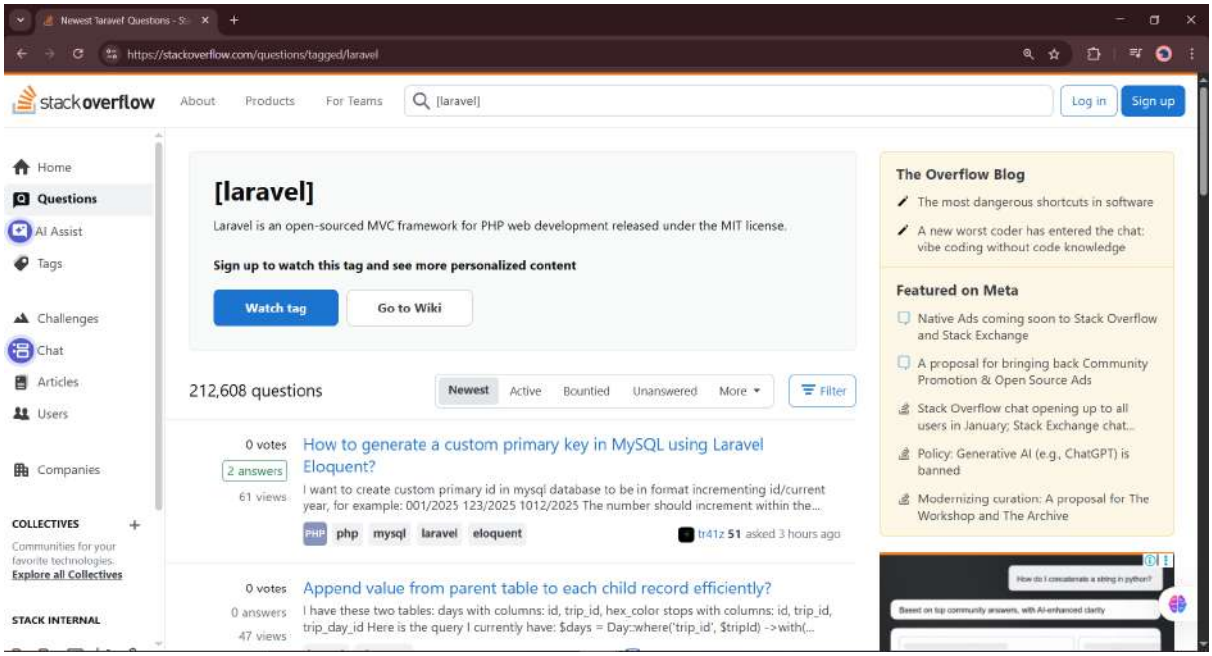
Skupnost in učenje

Tabela 2: Statistična primerjava virov in podpore za ogrodji Laravel in Yii

Vir/platforma	Laravel	Yii
Stack Overflow	~ 200.000 vprašanj	~ 40.000 vprašanj
GitHub repozitorij	> 30.000 zvezdic	~14.000 zvezdic
Uradna dokumentacija	Zelo obsežna	Tehnično natančna
Video tečaji	Laracasts (~ 2000 videov)	Omejeno
Forumi/Discord	Zelo aktivni	Manj aktivni

Pogostost posodobitev	Letne izdaje	Počasnejši razvoj

Vir: lasten, na podlagi analize uradnih spletnih strani in GitHub repozitorijev (Laravel, 2025), (Framework, 2024), (Overflow, 2026)



Slika 1: Število vprašanj, povezanih z ogrodjem Laravel

Vir: Stack Overflow (stanje na dan 5. 1. 2026)

3 VARNOST SPLETNIH APLIKACIJ

3.1 Uvod v varnost spletnih aplikacij

Varnost spletnih aplikacij je ključna komponenta sodobnega razvoja programske opreme. Količina občutljivih podatkov, ki jih obdelujejo spletne platforme, nenehno narašča zaradi široke uporabe interneta, spletnih storitev in e-trgovine (Stuttard.D and Pinto, 2011). Med najpogostejše varnostne grožnje sodijo vrivanje SQL, XSS in CSRF napadi (OWASP Foundation, 2023).

Zagotavljanje razpoložljivosti, zaupnosti in integritete podatkov je glavni cilj varnega razvoja spletnih aplikacij. Razvijalci bi morali že v zgodnji fazi razvoja vključiti varnostne ukrepe, ki zmanjšujejo verjetnost zlorabe. Osnovni varnostni ukrepi so na voljo v PHP, vendar je njihova učinkovita implementacija v veliki meri odvisna od znanja razvijalca. Ogrodja PHP, kot sta Laravel in Yii, zato postajajo bistven sestavni del sodobnega pristopa k varnemu razvoju.

3.2 Varnost v PHP

Čeprav PHP ne vključuje celovitega varnostnega sistema, razvijalcem omogoča uporabo temeljnih varnostnih ukrepov. PHP ponuja osnovne varnostne mehanizme (Group, PHP Group, 2024), vendar je njihova učinkovitost odvisna od znanja razvijalca. Med najpogostejšimi grožnjami so vrivanje SQL, XSS in CSRF napadi (Foundation O. , 2023). Najpogostejši mehanizmi so:

- validacija in filtriranje vnosov – preprečuje vdor škodljive kode v obrazce in parametre URL-jev;
- upravljanje sej – ščiti pred krajo sej in omogoča varno preverjanje pristnosti uporabnikov;
- komunikacija podatkov med strežnikom in odjemalcem je zaščitena, ko se uporablja HTTPS (SSL/TLS);
- uporaba pripravljenih stavkov v povezavi s knjižnicama PDO ali MySQLi preprečuje vrivanje SQL ukazov.

Ker mnogi razvijalci PHP uporabljajo lastne rešitve brez upoštevanja ustaljenih varnostnih načel, kljub tem zmogljivostim še vedno obstaja tveganje. V pomoč so lahko ogrodja, ki imajo vnaprej nameščene in preizkušene določene varnostne zaščite (Group, PHP Group, 2024).

3.3 Varnost v ogrodju Laravel

Med ogrodji PHP se Laravel ponaša z enim najobsežnejših varnostnih mehanizmov. Jedrna arhitektura ogrodja samodejno vključuje več ravni zaščite, ki tvorijo osnovo njegove varnosti:

1. Zaščita pred vrivanjem SQL: Laravelov Eloquent ORM in graditelj poizvedb samodejno uporablja vezane parametre. To pomeni, da so podatki varno obdelani, preden vstopijo v bazo podatkov.

2. Zaščita pred CSRF (Cross-Site Request Forgery): Vsak Laravelov obrazec je opremljen z žetonom CSRF, ki potrjuje legitimnost zahteve. Laravel zavrne zahtevo, če je žeton neveljaven ali manjka.

3. Zaščita pred XSS (Cross-Site Scripting): Laravel uporablja predloge Blade, ki se samodejno »izognejo« vsem izhodnim podatkom, da preprečijo dodajanje nevarne kode JavaScript na stran.

4. Avtentikacija in avtorizacija: Zgoščevanje gesel, dvofaktorska avtentikacija in zaščita poti so del Laravelovega integriranega sistema za avtentikacijo uporabnikov (Laravel Breeze, Fortify, Jetstream).

5. Šifriranje podatkov: Laravel šifrira občutljive podatke z uporabo algoritma AES-256 in knjižnice OpenSSL.

6. Upravljanje sej in gesel: Gesla so shranjena z algoritmom bcrypt ali Argon2, vgrajen mehanizem za regeneracijo sej ob prijavi uporabnika pa zmanjšuje tveganje ugrabitve sej.

Poleg teh funkcij Laravel podpira tudi varnostno testiranje z zunanjimi orodji, kot je Laravel Security Checker, ki išče ranljivosti in odvisnosti paketov.

3.4 Varnost v ogrodju Yii

Ogrodje Yii je znano po svoji strogi implementaciji varnostnih mehanizmov in stabilnosti. Ima nekoliko manj samodejnih funkcij kot Laravel, vendar omogoča razvijalcem natančen nadzor nad vsako varnostno plastjo.

Yiijeve glavne varnostne funkcije:

1. Preverjanje in filtriranje podatkov: Yii preverja tipe, dolžine, e-poštne naslove, številske vrednosti in druge podatke z uporabo kompleksnega nabora pravil v modelih.

2. Zaščita XSS in CSRF: Ogrodje uporablja funkcijo Html(HyperText Markup Language) ::encode() za samodejno obrambo pred napadi XSS. Poleg tega je zaščita CSRF samodejno vključena v vse obrazce z generiranjem žetonov.

3. Zaščita pred vrivanjem SQL: Yii ima lastna Active Record in Query Builder, ki samodejno vežeta parametre, da preprečita vnos škodljive kode SQL.

4. Upravljanje sej in preverjanja pristnosti: Yiijev vgrajeni sistem za preverjanje pristnosti in avtorizacije uporabnikov omogoča uporabo žetonov za API-je in shranjevanje sej v zbirki podatkov.

5. Šifriranje podatkov: Ogrodje ponuja razrede za varno upravljanje ključev in dovoljuje uporabo kriptografskih tehnik PHP, kot so hash(), password_hash() in openssl_encrypt().

6. Varnostne razširitve: Yii podpira različne razširitve, kot so yii2-security, yii2-authclient in yii2-user-management, ki izboljšujejo varnostno ogrodje.

Yii ima visoko stopnjo stabilnosti in predvidljivosti zaradi kombinacije teh tehnik, kar je ključnega pomena v večjih sistemih, kjer je treba varnost prilagoditi zahtevam naročnika (Framework, 2024).

3.5 Primerjava varnostnih mehanizmov

Tabela 3: Primerjava varnostnih mehanizmov med Laravel in Yii

Varnostni vidik	Laravel	Yii
Zaščita pred vrivanjem SQL	Samodejna z uporabo Eloquent ORM (Object-Relational Mapping)	Samodejna z uporabo Active Record
CSRF zaščita	Privzeto vključena v vse obrazce	Privzeto vključena v vse obrazce
XSS zaščita	Samodejna z Blade templatingom	Ročno s funkcijo
Šifriranje podatkov	AES-256	OpenSSL
Avtentikacija uporabnikov	Vgrajena (Breeze, Jetstream, Fortify)	Vgrajena, razširljiva preko modulov

Upravljanje sej	Avtomatska regeneracija sej	Podpora za shranjevanje sej v podatkovno bazo
Varnost razširitve	Obsežen ekosistem paketov	Manjši, a stabilen nabor razširitev
Zgoščevanje gesel	bcrypt, Argon2	bcrypt

Vir: lastna analiza dokumentacije (Laravel Documentation, n.d.), (Yii Documentation, n.d.)

Primerjava razkriva, da ponujata obe ogrodji visoko stopnjo varnosti, pri čemer Yii omogoča bolj tehnično in prilagodljivo upravljanje varnostnih nastavitev, medtem ko ponuja Laravel večjo avtomatizacijo in uporabniku prijaznejši pristop. Ena najpomembnejših faz razvoja je varnost spletnih aplikacij, ki je ne smemo prepustiti naključju. Številne varnostne funkcije, ki jih ponujata Yii in Laravel, močno zmanjšajo verjetnost ranljivosti in napadov. Zaradi avtomatizacije in uporabniku prijaznega Laravela lahko razvijalci delajo hitro in varno brez potrebe po dodatnih nastavitvah. Yii pa ponuja bolj organizirano in prilagodljivo metodo, ki je primerna za večje in bolj zapletene projekte z dobro definiranimi varnostnimi protokoli.

Na podlagi podatkov lahko ugotovimo, da je Yii bolj primeren za varnostno intenzivne aplikacije, ki potrebujejo stabilnost in natančen nadzor, medtem ko je Laravel boljša možnost za projekte, kjer sta hitrost in uporabnost glavni prednosti.

3.6 Vprašanja varstva podatkov in varnosti v spletnih zdravstvenih aplikacijah

Ker spletne aplikacije, ki obdelujejo izjemno občutljive osebne podatke, predstavljajo edinstveno varnostno tveganje (WHO, 2021), lahko uporabniki utrpijo hude posledice zaradi zlorabe ali razkritja podatkov, kot so diskriminacija, izguba zasebnosti ali psihološka škoda. Prav zato je treba k varnosti v tovrstnih aplikacijah pristopiti metodično in celovito, vključno z organizacijskimi in tehničnimi vidiki varstva podatkov. Za lažjo komunikacijo med uporabniki in prehranskimi strokovnjaki program NutriCare obdeluje osebne podatke, podatke o prehranjevalnih navadah in morebiti zdravstvene podatke. Zaradi tega je ena najpomembnejših značilnosti sistema varnost, ki neposredno vpliva na zaupanje uporabnikov in upoštevanje zakonskih zahtev.

Podrobnosti o varnosti pri obdelavi zdravstvenih podatkov

Ker zdravstveni podatki zagotavljajo informacije o zdravju, življenjskem slogu in morebitnih nevarnostih osebe, veljajo za posebej zaščitene osebne podatke. To se nanaša predvsem na

informacije o prehrani, telesni teži, ciljih uporabnikov in strokovni komunikaciji v kontekstu aplikacije NutriCare. Njihova narava je občutljiva in zahteva ustrezno stopnjo zaščite, čeprav ne gre nujno za klinične zdravstvene podatke.

Zagotavljanje zaupnosti – torej omejevanje dostopa do podatkov na tiste, ki so za to pooblaščeni, je eno glavnih vprašanj pri obdelavi takšnih podatkov (Sommerville, Software engineering, 2016). Poleg tajnosti sta bistveni tudi razpoložljivost, ki uporabnikom omogoča dostop do sistema, ko ga potrebujejo, in celovitost podatkov, ki zagotavlja, da podatki niso bili spremenjeni. Triada CIA – zaupnost, celovitost in razpoložljivost – predstavlja temeljne ideje informacijske varnosti.

Tipična tveganja za varnost spletnih aplikacij

Najpogostejša varnostna tveganja spletnih aplikacij vključujejo vrivanje zlonamerne kode, krajo sej in neustrezno upravljanje preverjanja pristnosti, kar izpostavlja tudi OWASP (Foundation O. , 2023). Slabo upravljanje preverjanja pristnosti, nezadostna uporabniška dovoljenja, nepravilno ravnanje z gesli in nezadostna varnost podatkov v podatkovnih bazah so najpogostejša tveganja, ki jih poudarja tudi OWASP. Nezadostno varen sistem v kontekstu aplikacije NutriCare lahko uporabnikom omogoči dostop do njihovih zasebnih podatkov ali razkrije pogovore s strokovnjaki za prehrano. Takšni dogodki lahko negativno vplivajo na zaupanje uporabnikov in pravno odgovornost skrbnika sistema. Zato je ključnega pomena upoštevati varnostne koncepte, kot so najmanj potrebne pravice, ločevanje uporabniških vlog ter pogosto testiranje in preverjanje varnostnih mehanizmov, ki so že v fazi razvoja.

Nadzor dostopa, avtorizacija in preverjanje pristnosti

Ustrezno upravljanje uporabniške identitete je ena najpomembnejših komponent varnosti spletnih aplikacij. Medtem ko dovoljenja nadzorujejo funkcije in podatke, do katerih lahko vsak uporabnik dostopa, preverjanje pristnosti potrjuje, da je uporabnik tisti, za katerega se izdaja.

Smiselno je razlikovati med različnimi uporabniškimi vlogami v programu NutriCare, kot so sistemski skrbniki, nutricionisti in običajni uporabniki. Da bi zmanjšali možnost zlorabe in nenamernih napak, mora imeti vsako delovno mesto jasno opredeljene pravice dostopa. Poleg tega so priporočljive tehnike varnega shranjevanja gesel, kot so enosmerne zgoščevalne funkcije z uporabo soli in obramba pred napadi s surovo silo. Uporaba večfaktorske avtentikacije, ki poleg gesla zahteva dodatno preverjanje identitete uporabnika, na primer prek mobilne naprave ali enkratnega gesla, bi lahko zagotovila dodatno stopnjo varnosti.

Varnost shranjevanja in prenosa podatkov

Zaščita podatkov med prenosom in shranjevanjem je še ena ključna komponenta varnosti. Prestrezanje podatkov med prenosom se prepreči z uporabo protokola HTTPS, ki zagotavlja šifrirano povezavo med odjemalcem in strežnikom, vendar pri obdelavi občutljivih podatkov pogosto ne predstavlja zadostne zaščite (Smith, 2016). Podatki v zbirki podatkov morajo biti zadostno zaščiteni, kar vključuje pogoste varnostne kopije, omejen dostop do zbirke podatkov in šifriranje pomembnih polj. Posebno pozornost je treba nameniti tudi varnosti varnostnih kopij, saj lahko kopije, ki niso dovolj zaščitene, predstavljajo dodatno nevarnost.

Sledi spremljanja in beleženje dogodkov

Beleženje dogodkov omogoča sledljivost dostopov in je ključno za odkrivanje varnostnih incidentov ter zagotavljanje skladnosti s standardi informacijske varnosti (International Organization for Standardization, 2013). Sledi spremljanja zagotavljajo pregled nad tem, kdo in kdaj je dostopal do določenih podatkov, kar je ključnega pomena z vidika varnosti in skladnosti s predpisi. Beleženje dostopov bi olajšalo obvladovanje morebitnih nesreč in omogočilo hitrejše odkrivanje nepravilnosti v programu NutriCare. Ta metoda naredi sistem bolj pregleden, uporabniki z večjimi privilegiji pa so bolj odgovorni.

Pomen varnosti za zaupanje uporabnikov

Zaupanje uporabnikov je neposredno odvisno od varnosti spletne aplikacije. Zaupanje je bistvenega pomena za dolgoročno uporabo aplikacije NutriCare, kjer si uporabniki izmenjujejo prehranske in osebne podatke. Če so uporabniki prepričani, da so njihovi podatki dovolj zaščiteni in da se z njimi ravna v skladu z veljavno zakonodajo, bodo bolj nagnjeni k uporabi programa. Temelj za uspešno in dolgotrajno delovanje aplikacije v resničnem okolju je celovit pristop k varnosti, ki vključuje organizacijske ukrepe, tehnične rešitve in nenehne izboljšave.

Načrtovanje varnosti in metoda »varnost po zasnovi«

Načelo, imenovano »varnost po zasnovi« (Security by design), ki poudarja vključitev varnostnih vidikov že od samega začetka razvoja programske opreme, postaja vse pomembnejše pri ustvarjanju sodobnih spletnih aplikacij (European Union, 2016). Ta metoda pomeni, da so varnostne zahteve del temeljne arhitekture sistema, namesto da se varnostne metode uvajajo pozneje. Takšen razvoj zmanjšuje možnost varnostnih napak in sčasoma povečuje stabilnost in zanesljivost aplikacije. Glede na to, da program NutriCare obdeluje osebne in morebiti zdravstvene podatke uporabnikov, je pristop varnosti že z zasnovo še

posebej pomemben, zato je v fazi načrtovanja podatkovnega modela bistveno upoštevati, ali so podatki za delovanje sistema dejansko potrebni, in preprečiti nepotrebno pridobivanje občutljivih podatkov. Poleg neposrednega zmanjševanja varnostnih tveganj je načelo minimizacije podatkov (data minimization) skladno s predpisi GDPR. Segmentacija sistema in delitev nalog med njegovimi različnimi komponentami sta prav tako ključna vidika načrtovanja varnosti. Ločitev uporabniškega vmesnika, strežniške logike in podatkovne plasti omogoča boljši nadzor dostopa ter enostavnejše prepoznavanje in odpravljanje morebitnih varnostnih pomanjkljivosti. Poleg tega takšna arhitektura omogoča enostavnejšo nadgradnjo določenih sistemskih komponent, ne da bi to vplivalo na aplikacijo kot celoto.

Upravljanje in odzivanje na varnostne incidente

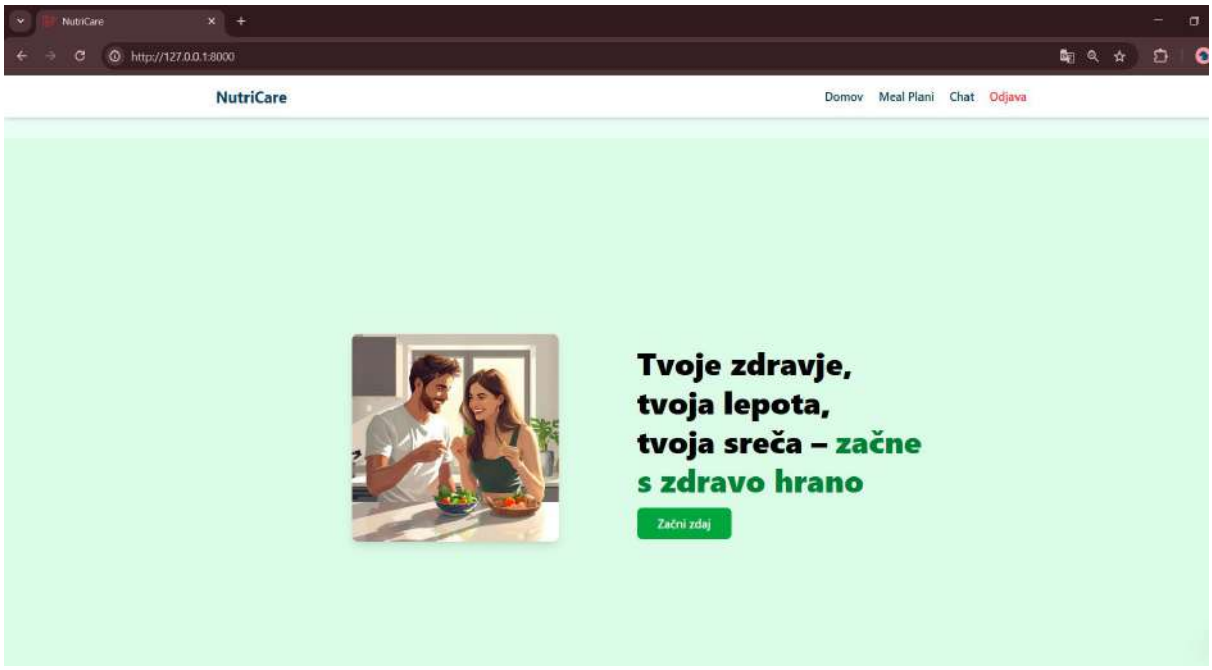
Ključnega pomena je, da je aplikacija pripravljena na morebitne varnostne incidente, saj ni popolne zaščite sistema, niti z ustreznimi varnostnimi ukrepi (NIST, 2012). Prepoznavanje, ocenjevanje in odzivanje na varnostne incidente ter odpravljanje njihovih posledic so del upravljanja varnostnih incidentov. Smiselno bi bilo uvesti preprost mehanizem za prepoznavanje nepravilnih dejavnosti v aplikaciji NutriCare, kot so ponavljajoči se neuspešni poskusi prijave ali nenavadni vzorci dostopa do podatkov. Te metode zmanjšujejo verjetnost večje škode in omogočajo hitrejše odkrivanje morebitnih napadov.

Vloga uporabnikov pri ohranjanju varnosti

Vedenje uporabnikov je za varnost spletne aplikacije prav tako pomembno kot tehnične rešitve. Ozaveščanje uporabnikov o varnostnih tveganjih pomembno prispeva k zmanjšanju človeškega faktorja kot vira ranljivosti sistema (Foundation O. , 2023), zato je smiselno uporabnike izobraževati o temeljnih varnostnih smernicah, kot so ustvarjanje močnih gesel, previdno deljenje dostopnih podatkov in redno posodabljanje osebnih gesel. Izvedljivo bi bilo uporabnikom v aplikaciji NutriCare zagotoviti kratka varnostna opozorila ali predloge, ki bi spodbujali varnejšo uporabo sistema. Ta metoda prispeva k izboljšanju varnosti aplikacije in poveča zaupanje uporabnikov vanjo.

4. RAZVOJ APLIKACIJE V PHP OGRODJIH

4.1 Razvoj v ogrodju Laravel



Slika 2: Razvoj aplikaciji NutriCare v ogrodju Laravel

Vir: lasten

4.1.1 Priprava razvojnega okolja

Aplikacijo NutriCare smo ustvarili z uporabo ogrodja Laravel različice v12.41.1.

Razvojno okolje ima sestavo:

- PHP 8.3,
- Composer,
- podatkovna baza MySQL,
- Artisan CLI za Laravel,
- Laravel Breeze za preverjanje.

Za namestitev Laravela je bil uporabljen ukaz: composer create-project laravel/laravel nutricare (Composer, Composer, 2024).

4.1.2 Tehnologije v NutriCare

Tabela 4: Tehnologije v NutriCare

Tehnologija	**Namen v projektu NutriCare**	**Tip licence**	**Cena**
PHP 8.3	Strežniški programski jezik	Open Source (PHP License)	0 €
Laravel 12	Backedn ogrodje (MVC, varnost)	MIT (Open Source)	0 €
Yii2	Alternativno PHP ogrodje	BSD License	0 €
MySQL	Podatkovna baza	GPL	0 €
Composer	Upravljanje odvisnosti	Open Source	0 €
Laravel Breeze	Avtentikacija	MIT	0 €
Bootstrap 5	UI oblikovanje	MIT	0 €
JavaScript/AJAX	Asinhrona komunikacija	Open Source	0 €
GitHub	Verzijsko sledenje kode	Freemium	0 €
phpMyAdmin	Upravljanje baze	GPL	0 €

Vir: lasten, 2025

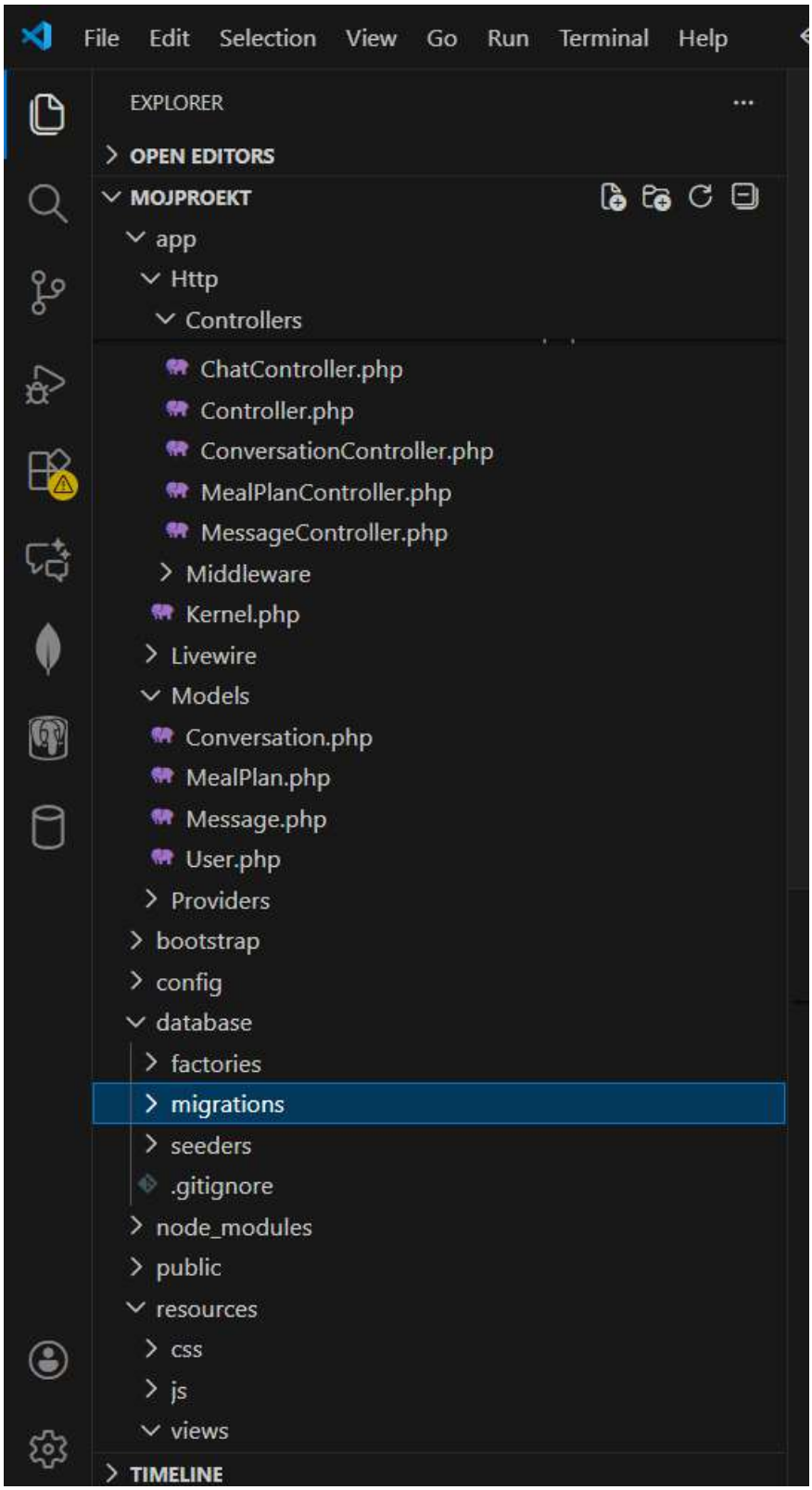
Vse tehnologije, uporabljene v projektu NutriCare, so brezplačne in odprtokodne, kar potrjuje cenovno dostopnost projekta in primernost za uporabo v zagonskih in izobraževalnih okoljih.

Vir: Sestavila avtorica na podlagi uradnih dokumentacij (Corporation, 2024), (Team, 2024) (Group, PHP Group, 2024).

4.1.3 Struktura projekta

Laravel uporablja definirano MVC arhitekturo:

- modeli (app/Models) za upravljanje podatkov,
- kontrolerji (app/Http/Controllers/) za poslovno logiko,
- pogledi (resources/views/) – Blade predloge,
- migracije (database/migrations/) za upravljanje baze.



Slika 3: Prikaz MVC arhitekture

Vir: lasten

4.1.4 Implementirane funkcionalnosti

NutriCare je spletna aplikacija, ki povezuje uporabnike s prehranskimi svetovalci, ustvarjena pa je bila z ogrodjem Laravel. Zanesljivost in varnost v spletnem okolju omogoča arhitektura aplikacije, ki daje velik poudarek funkcionalnosti in varnosti.

Avtentikacija in registracija uporabnikov

Večina sodobnih spletnih aplikacij je zgrajena na mehanizmu prijave in registracije. Ta sistem, ki ponuja preprost in učinkovit način upravljanja uporabnikov, je bil implementiran z modulom Laravel Breeze. Sistem ima vse bistvene funkcije, vključno s preverjanjem e-poštnega naslova, obnovitvijo gesla, prijavo, odjavo in registracijo novega uporabnika. Varnost gesel je bila deležna posebne pozornosti. Ker Laravel zgoščuje gesla z algoritmom bcrypt, so shranjena v nepovratni obliki. Geslo se nikoli ne shrani ali obdela v berljivi obliki – namesto tega se primerja z zgoščeno vrednostjo v bazi podatkov, ko se uporabnik prijavi. Napade ponarejanja zahtev preprečuje žeton CSRF, ki ščiti vsak obrazec v aplikaciji.

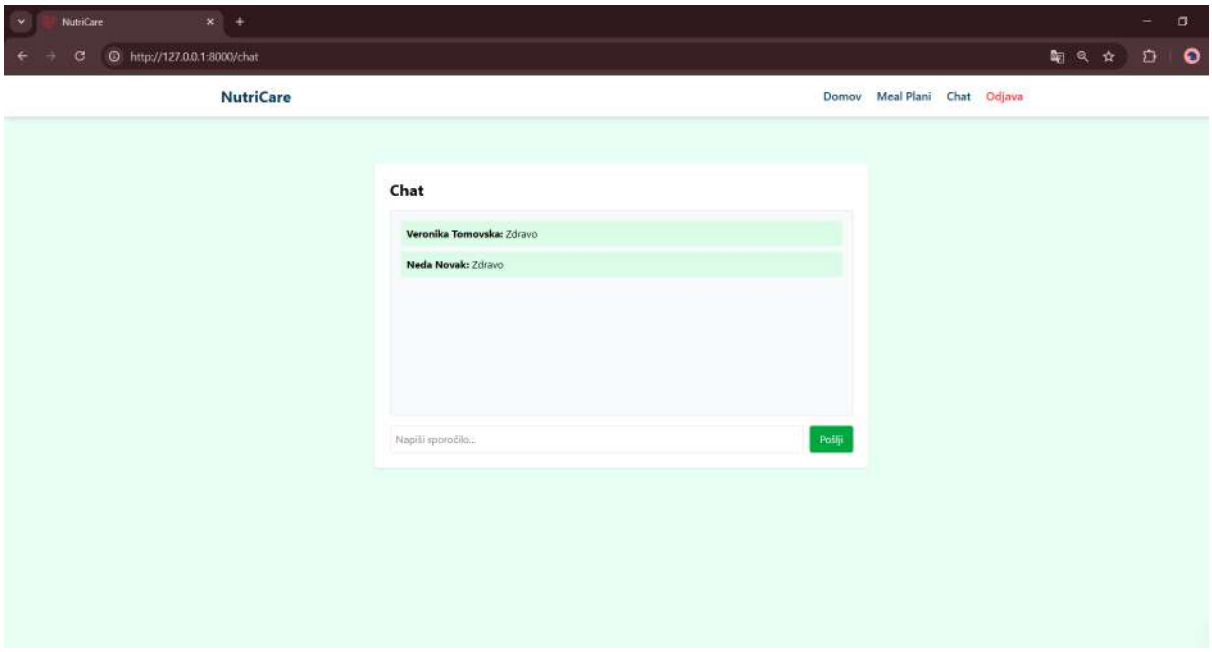
Sistem uporabniških vlog

Program razlikuje med dvema kategorijama uporabnikov: redni uporabniki in prehranski svetovalci. Za izvedbo te razlike se uporablja polje vloge v tabeli uporabnikov, ki dovoljuje različne pravice dostopa in funkcije za vsako vrsto uporabnika.

Za nadzor dostopa je bila ustvarjena vmesna programska oprema po meri, ki preveri vlogo uporabnika, preden mu dodeli dostop do kateregakoli omejenega območja programa. Ta metoda zagotavlja, da lahko redni uporabniki dostopajo do funkcij, ki so jim namenjene, medtem ko lahko svetovalci dostopajo le do funkcij, ki so namenjene njihovemu delovnemu mestu.

Komunikacijski modul

Komunikacijski modul ki omogoča dvosmerno komunikacijo med uporabniki in svetovalci, je ena bistvenih komponent aplikacije. Sistem je zasnovan kot enostaven vmesnik za pošiljanje in prejemanje sporočil, vsaka interakcija pa je dokumentirana v organizirani bazi podatkov.



Slika 4: Prikaz pogovora z aplikacijo NutriCare

Vir: lasten

Varnostni postopki

Skozi celoten razvojni proces je bila varnost izjemnega pomena. Laravel ponuja širok nabor integriranih varnostnih mehanizmov, ki so bili metodično implementirani in preverjeni.

Zaščita pred CSRF napadi

Napadi CSRF, pri katerih napadalec poskuša uporabnika prisiliti k izvedbi neželenih dejanj na spletnem mestu, ko je prijavljen, predstavljajo znatno varnostno tveganje. Ob vzpostavitvi uporabniške seje Laravel samodejno generira edinstven CSRF žeton, ki je shranjen v seji. Ta žeton mora biti vključen v vsak obrazec, ki pošilja zahtevo strežniku. Middleware nato preveri, ali se žeton iz seje ujema z žetonom, poslanim v zahtevi. Če se žetona ujemata, je zahteva sprejeta, sicer je zavrnjena.

Zaščita pred XSS napadi

Napadi XSS omogočajo napadalcem vstavljanje zlonamerne kode JavaScript v spletno stran, kar lahko povzroči ugrabitev uporabniške seje ali krajo podatkov. Laravel zagotavlja zaščito pred XSS prek predlognega mehanizma Blade, ki samodejno »escapira« izpisane spremenljivke, kadar se uporablja zapis {{ }}. V tem primeru se potencialno nevarni znaki

pretvorijo v varne HTML entitete. Če pa razvijalec uporabi zapis {!! !!}, se vsebina izpiše brez »escapiranja«, kar lahko predstavlja varnostno tveganje.

Zaščita pred vrivanjem SQL

Z vstavljanjem zlonamerne kode SQL v poizvedbe lahko napadalec pridobi nepooblaščen dostop do podatkov s tehniko, znano kot vrivanje SQL (injekcija). Tovrstni napad popolnoma preprečita Laravel Eloquent ORM in Query Builder, ki samodejno uporabljata vezane parametre (pripravljene stavke). Zlonamerne kode ni mogoče vstaviti, ker se podatki vedno obravnavajo kot vrednosti in ne kot komponenta stavkov SQL.

Avtentikacija in avtorizacija

Komponente vmesne programske opreme podpirajo sistem preverjanja pristnosti tako, da potrdijo identiteto uporabnika, preden mu odobrijo dostop do zaščitenih virov. Natančen nadzor dostopa je omogočen tudi z implementacijo razredov politik (angl. *policies*) za podrobno avtorizacijo. Politika sporočil na primer zagotavlja, da si uporabnik lahko ogleda le sporočila, ki jih je poslal sam ali so namenjena njemu.

Zaščita pred Mass Assignment

Ko napadalec poskuša spremeniti polja v zbirki podatkov, ki ne bi smela biti dostopna, se to imenuje Mass Assignment (napad množične dodelitve). Laravel se temu izogne z uporabo atributov $fillable ali $guarded v modelih, ki natančno določajo, katera polja lahko uporabnik spremeni.

Rate Limiting

Omejitev hitrosti, ki omejuje število zahtev, ki jih lahko uporabnik opravi v določenem časovnem obdobju, se uporablja za zaščito pred napadi DDoS in zlorabo sistema. To zagotavlja stabilnost sistema in preprečuje avtomatizirane napade.

4.1.5 Prednosti in izzivi v Laravelu

Pri ustvarjanju spletne aplikacije NutriCare smo imeli priložnost temeljito preizkusiti ogrodje Laravel in odkriti več prednosti, ki močno izboljšajo učinkovitost in zanesljivost ustvarjanja sodobnih spletnih aplikacij. Hkrati smo naleteli na nekaj težav, ki jih je treba upoštevati pri načrtovanju arhitekture in dolgoročnem vzdrževanju kode.

Prednosti ogrodja Laravel

Hitra namestitev z obsežnim ekosistemom vnaprej nameščenih paketov

Ena glavnih prednosti Laravela je njegov neverjetno velik ekosistem paketov, ki omogoča hiter razvoj zapletenih zmogljivosti brez nepotrebnega ponovnega izvajanja temeljnih procesov. Med najbolj koristnimi so:

- Laravel Breeze in Jetstream – v samo nekaj minutah omogočata popolno nastavitev sistema za preverjanje pristnosti, vključno z registracijo, prijavo, obnovitvijo gesla in preverjanjem e-pošte.
- Laravel Breeze za preprosto in varno avtentikacijo API.
- Laravel Horizon za bolj kompleksno upravljanje čakalnih vrst Redis.
- Laravel Telescope – orodje za odpravljanje napak in spremljanje podatkov aplikacij.

Celotna implementacija sistema za preverjanje pristnosti projekta NutriCare z uporabo paketa Breeze je bila končana v približno 30 minutah. Ta primer zelo jasno prikazuje, kako Laravel spodbuja hitre, organizirane in varne metode razvoja.

Odlična in temeljita dokumentacija

Laravel izstopa po svoji uradni dokumentaciji, ki je razporejena po ločenih funkcionalnih enotah in je neverjetno dobro strukturirana ter se redno posodablja. Jasne razlage, primeri in temeljite reference v dokumentaciji močno olajšajo učenje ogrodja.

Razvijalci imajo poleg dokumentacije dostop do številnih drugih virov, vključno z:

- Laracasts, ki ponuja skoraj 2000 video tečajev z navodili;
- Laravel News, ki pogosto objavlja najboljše prakse in nove funkcije;
- Laravel Daily, ki vsak dan ponuja koristne napotke.

Razpoložljivost tako obsežnega znanja omogoča hitrejše reševanje težav in boljšo kakovost kode.

Sistem predlog Blade, ki je enostaven za uporabo

Z Laravelovo tehnologijo predlog Blade lahko pišemo jasno, jedrnato in berljivo kodo. Njena razširljivost in intuitivnost pomagata ločiti predstavitev od logike, kar izboljša vzdrževanje programa. Blade dodatno izboljša strukturo uporabniškega vmesnika, saj olajša uporabo komponent, direktiv in dedovanja predlog.

Uporaba Artisan CLI za avtomatizacijo ponavljajočih se opravil

Z omogočanjem ustvarjanja modelov, migracij, krmilnikov in drugih komponent aplikacije z enostavnimi ukazi Laravelovo orodje ukazne vrstice Artisan dramatično poveča produktivnost. Za razvijalce slednje prihrani veliko časa, ki bi ga sicer porabili za monotono delo. Glede na stopnjo razvoja ugotavljamo, da uporaba Artisana v povprečju prihrani dve do tri ure na teden.

Odzivna in aktivna skupnost

Med ogrodji PHP se Laravel ponaša z eno največjih in najbolj aktivnih skupnosti. Pomoč je dostopna na:

- Discordu za Laravel (več kot 50.000 aktivnih uporabnikov),
- uradnem forumu za Laravel,
- Stack Overflow (več kot 200.000 vprašanj, povezanih z Laravelom),
- GitHub Discussions, kjer se lahko obrnemo tudi na glavne razvijalce ogrodja.

Ker je skupnost tako velika, je pomoč zlahka na voljo – pogosto v nekaj minutah – kar močno pomaga pri odpravljanju težav in prepoznavanju najboljših praks.

Enostavno delo z bazami podatkov in eleganten ORM

Delo z bazami podatkov na objektno usmerjen način omogoča Eloquent ORM, ki ponuja tudi preprost in razumljiv jezik za poizvedbe. Modeli, relacije in tako imenovani »graditelj poizvedb« zmanjšujejo verjetnost napak, izboljšujejo berljivost kode in pospešujejo razvoj.

Integrirana pomoč pri testiranju

Laravel v celoti podpira testiranje enot in funkcij. To omogoča razvijalcem, da skrbno preverijo delovanje posameznih funkcij in povečajo zanesljivost aplikacije, kar je bistveno za dolgoročno vzdrževane projekte, kot je NutriCare.

Slabosti in težave Laravela

Povečana poraba virov za bolj zapletene poizvedbe

Eloquent lahko pri kompleksnih poizvedbah povzroči tako imenovani problem N+1 poizvedb, kar vpliva na delovanje aplikacije, kljub temu da je neverjetno sodoben in preprost.

Rešitev je uporaba takojšnjega nalaganja s funkcijo with() in orodij, kot sta Laravel Debugbar ali Laravel Telescope, za prepoznavanje neučinkovitih poizvedb.

Težavnost koncepta vsebnika storitev

Vbrizgavanje odvisnosti in vsebniki storitev zagotavljajo temelj velikega dela Laravela. Čeprav ta metoda ponuja odlično preizkušanje in prilagodljivost, je za začetnike kompleksnejša. Običajno traja dva do tri tedne intenzivnega učenja, da se ideja v celoti razume, preden jo je mogoče uspešno uporabiti.

Kompleksnost naprednih funkcij

Za funkcije, kot so sistem dogodkov, čakalne vrste in oddajanje dogodkov, sta potrebna dodatna konfiguracija in poznavanje več komponent hkrati (Redis, spletna vtičnica itd.). Nastavitev in pravilno testiranje teh področij v projektu NutriCare je trajalo nekoliko dlje.

Kompleksnost sofisticiranih funkcij

Funkcije, kot so oddajanje, čakalne vrste in sistem dogodkov, zahtevajo dodatno konfiguracijo in sočasno razumevanje več komponent (Redis, spletna vtičnica itd.). V okviru razvoja aplikacije NutriCare je nastavitev in temeljito testiranje teh področij trajalo nekoliko dlje.

Redne posodobitve in potreba po pogostih različicah

Laravel redno izdaja nove glavne različice:

- Laravel 8 (leta 2020),
- Laravel 9 (leta 2022),
- Laravel 10 (leta 2023),
- Laravel 11 (leta 2024).

Slednje zahteva pravočasen odziv na spremembe ogrodja in pogoste posodobitve kode. To pomeni dodatno delo, ki ga je treba upoštevati pri načrtovanju življenjskega cikla aplikacije za dolgoročne projekte.

Sorazmerno velika velikost osnovnega projekta

Osnovna namestitev ogrodja Laravel zahteva veliko več prostora na disku kot navadna aplikacija, razvita v čistem PHP, predvsem zaradi številnih vgrajenih paketov in odvisnosti. Tipične namestitve so sestavljene iz:

- mape vendor (odvisnosti Composerja), ki obsega približno 70 MB;
- mape node_modules, ki obsega približno 50 MB, če se v aplikaciji uporablja Vite ali Laravel Mix.

V primerjavi z enostavno aplikacijo, razvito v čistem PHP, Laravel zahteva več prostora na disku zaradi vključevanja številnih odvisnosti, knjižnic in podpornih komponent. Ta večja programska osnova je posledica obsežnega ekosistema in širokega nabora vgrajenih funkcionalnosti.

4.2 Razvoj z ogrodjem Yii

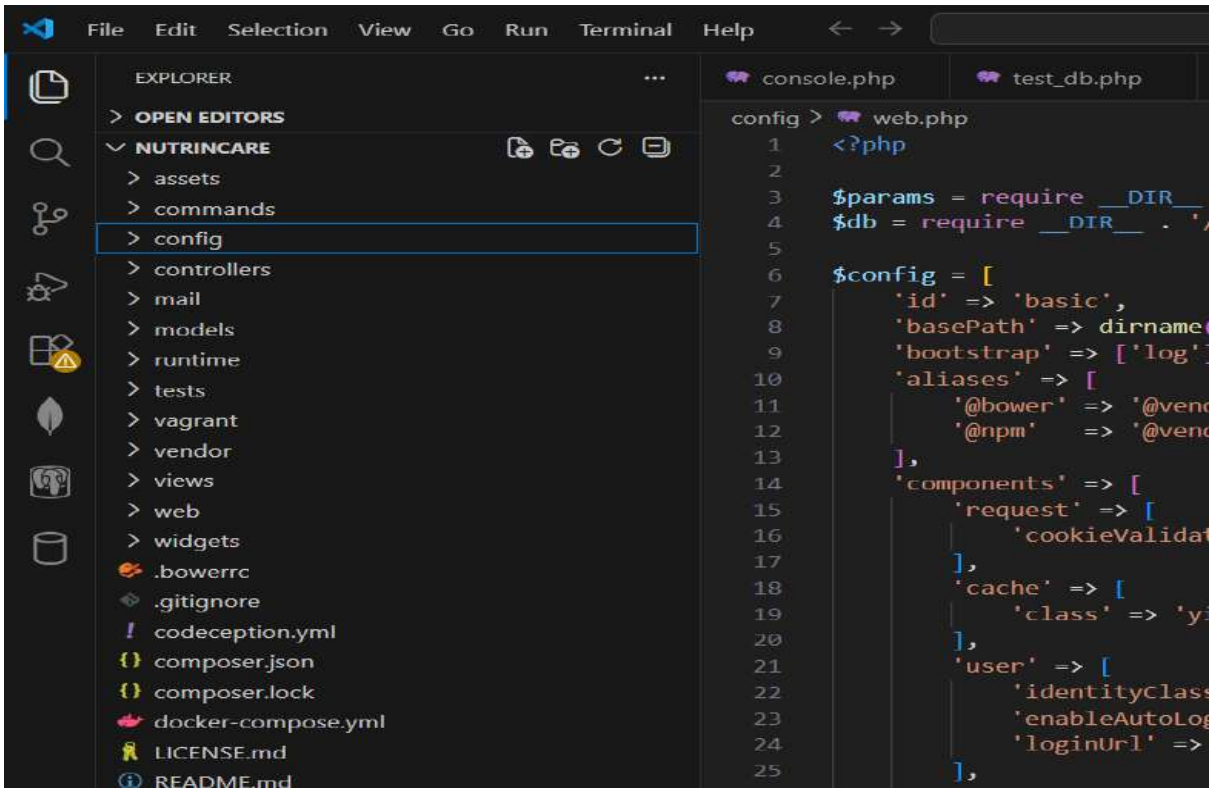
Za praktični del zaključnega dela smo v ogrodju Yii ustvarili enako spletno aplikacijo NutriCare, ki služi kot platforma za pomoč pri prehranskem svetovanju. Glavni cilj aplikacije je olajšati učinkovito digitalno komunikacijo in upravljanje prehranskih načrtov med dvema pomembnima skupinama:

- nutricionisti, ki aktivno komunicirajo s pacienti, ustvarjajo individualizirane prehranske režime in ocenjujejo njihove prehranjevalne navade;
- prejemniki prehranskih načrtov, ki spremljajo svoj razvoj in po potrebi poiščejo strokovni nasvet.

Razvijali smo jo z uporabo ogrodja PHP Yii2, saj omogoča hiter, organiziran in varen razvoj zapletenih spletnih aplikacij. Vzpostavili smo celovito sistemsko rešitev z uporabo MySQL in phpMyAdmin, ki jo sestavljajo:

- avtorizacija in preverjanje pristnosti uporabnikov,
- upravljanje uporabniških vlog,
- moduli za načrte obrokov,
- modul za klepet,
- intuitiven grafični uporabniški vmesnik,
- CRUD (funkcije upravljanja),
- zaščitni ukrepi.

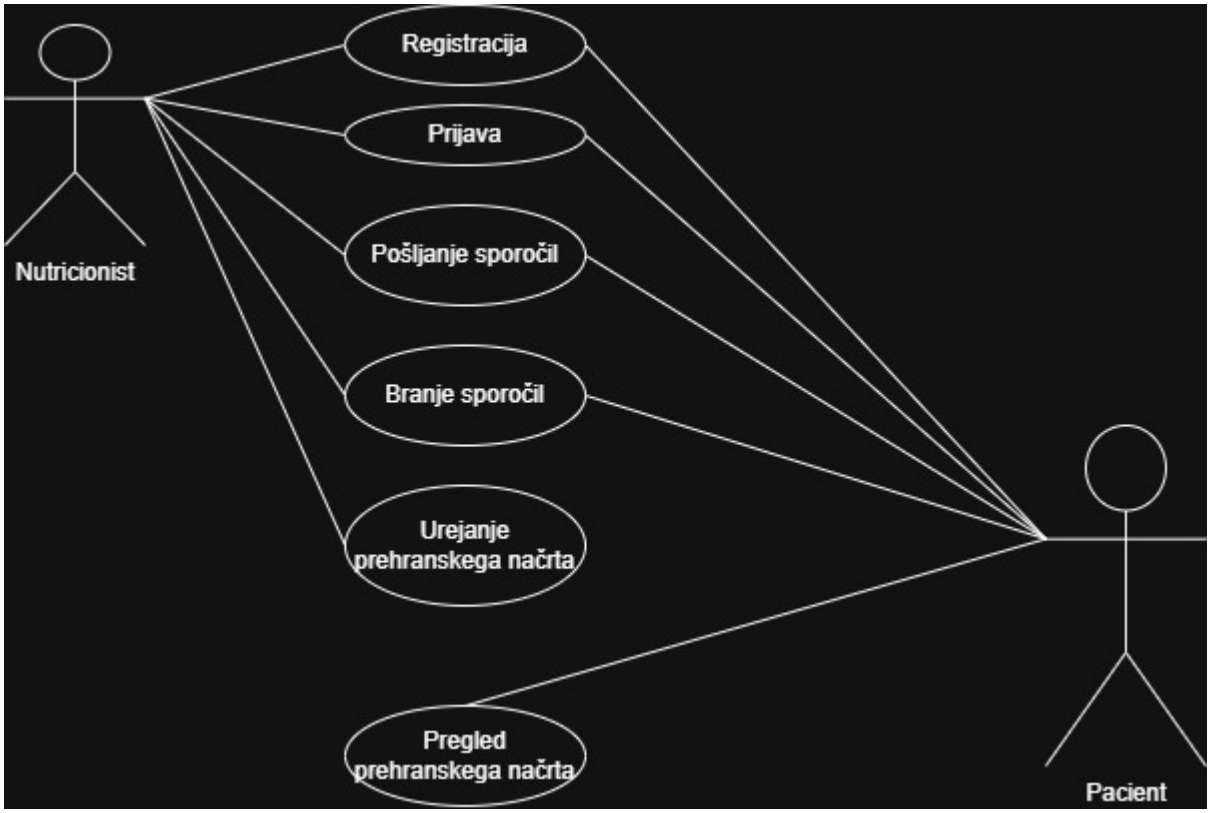
Aplikacija NutriCare vključuje najboljše prakse razvoja spletnih sistemov in je bila zgrajena z uporabo arhitekture MVC in sodobnih spletnih standardov.



Slika 5: Prikaz MVC arhitekture v ogrodju Yii

Vir: lasten

4.2.1 Sistemska arhitektura in uporaba tehnologije



Slika 6: Diagram primerov uporabe aplikacije NutriCare

Vir: lasten

Arhitektura MVC (Model–View–Controller), ki ločuje prikazno plast od poslovne logike in podatkovne plasti, je temelj aplikacije NutriCare. Pri implementaciji smo se odločili za uporabo ogrodja Yii2, ki vključuje številna orodja (ActiveRecord, Gii, komponente, filter AccessControl itd.) ter robusten in dosleden pristop MVC. Takšna arhitektura prispeva k boljši možnost ponovne uporabe, lažjemu testiranju in vzdrževanju kode.

4.2.2 Yii MVC arhitektura

Yii2 uporablja jasno delitev dela med MVC komponentami, pri čemer so vsaki ravni dodeljene različne funkcije. Posamezne komponente vzorca MVC so v nadaljevanju predstavljene na implementaciji aplikacije NutriCare.

Model (M)

Cilj modela je prikazati postopke in smernice, ki jih nadzorujejo. Podatkovna baza je tesno povezana tako z modelom kot z ActiveRecord.

Primarni modeli NutriCare so:

- model za upravljanje uporabnikov, uporabniških vlog, prijavne poverilnice, zgoščevanje gesel in priročne metode (kot je isNutritionist());
- MealPlan: model za načrte obrokov, ki vključuje spremenljivke, kot so breakfast/zajtrk, plan_date, nutritionist_id in patient_id;
- sporočilo: model za komunikacijo, ki vključuje spremenljivke sender_id, receiver_id, message, created_at in is_read;
- Yii2 ActiveRecord: z ActiveRecord lahko vsako tabelo hitro preslika v ustrezen razred PHP.

Prednosti:

Preslikava: Razred MealPlan v mapi app\models je preslikan v tabelo meal_plan.

To omogoča izvajanje objektno usmerjenih poizvedb, kot prikazuje naslednji primer:

```
$plans = MealPlan::find()
    ->where(['patient_id' => $patientId])
    ->orderBy(['plan_date' => SORT_DESC])
    ->all()
```

Poleg tega Yii ponuja scenarije (več nizov pravil za ustvarjanje ali posodobitev), kar je koristno pri delu z različnimi obrazci.

Pri operacijah, ki vključujejo več zaporednih zapisov ali sprememb v podatkovni bazi (npr. ustvarjanje načrta obrokov in obveščanje baze podatkov), smo uporabili transakcije.

Pogled (V)

V predstavitveni plasti se uporabljajo HTML, CSS in JavaScript. Pogledi so lahko implementirani kot tradicionalne predloge PHP, pri čemer se uporabljajo pomožni razredi, npr. HTML, ActiveForm in URL.

V NutriCare smo pripravili edinstvene poglede, ki so opisani spodaj.

- Za prijavo in registracijo smo uporabljali varni obrazec z ActiveForm, žetoni CSRF in preverjanje na strani strežnika.
- »Ustvarite, spremenite in si oglejte načrt obrokov« z uporabo odzivne postavitve (Bootstrap + CSS po meri), kartic in uporabniškega vmesnika z zeleno tematiko.
- Klepet (sporočilo): chat.php, delni _messages.php in _single_message.php; jQuery se uporablja na sprednjem delu za pošiljanje in anketiranje sporočil AJAX.

Svežnji sredstev in postavitev

Za običajne elemente (navigacijska vrstica, noga strani) Yii omogoča uporabo postavitev (kot je main.php). Za razporeditev sloga in skript smo uporabili svežnje sredstev (Bootstrap 5 + CSS po meri).

Delni elementi in gradniki

Za delne prikaze (kot so seznami sporočil) smo uporabili funkcijo renderPartial() in ločene delne datoteke, kar ohranja modularnost kode in omogoča učinkovitejše posodabljanje vsebine z uporabo AJAX zahtev.

Krmilnik (C)

Funkcije krmilnika vključujejo povezovanje modelov in pogledov ter upravljanje zahtev HTTP, avtorizacijo, preverjanje veljavnosti in preusmeritve.

V aplikaciji NutriCare je bil implementiran MealPlanController, ki vsebuje CRUD akcije: actionCreate(), actionUpdate(), actionView(), actionDelete().

Nadzor dostopa in filtri

Za zaščito končnih točk smo uporabili filter AccessControl.

MVC v Yii2 ponuja varnost in vzdrževanje, jasno delitev nalog in povečano produktivnost razvoja (z Gii in ActiveRecord). Pravzaprav nam je ta arhitektura omogočila, da smo ustvarili modularni program NutriCare, pri čemer je vsak modul (klepet, načrt obrokov) neodvisen, preizkušen in razširljiv. Yii2 ponuja dovolj orodij tako za uporabo v obsežni produkciji kot za hitro izdelavo prototipov.

Modul za načrt prehrane

Osrednja funkcionalnost NutriCare je modul za načrt prehrane, ki nutricionistu omogoča, da vsakemu pacientu zagotovi prilagojena in organizirana prehranska priporočila. Cilj modula je ponuditi strukturiran, pregleden in strokoven način sledenja prehrani in spreminjanja načrtov glede na razvoj vsakega posameznika.

Funkcionalnosti modula

V času zasnove in izvedbe smo zagotovili naslednje ključne vloge:

1. Izbira pacientov

Dietetiku je na voljo seznam uporabnikov, ki igrajo vlogo pacientov. Najprej izbere pacienta, za katerega bo nov prehranski načrt ustvarjen. To zagotavlja, da je vsak načrt neposredno povezan s posameznim uporabnikom.

2. Določanje datuma načrta prehrane

Vsaka strategija ima natančen dan. To omogoča sledenje prehranskim priporočilom skozi čas, lažje organiziranje načrtov in zgodovinski povzetek. Da bi se izognili zmedi in nasprotujočim si dogovorom, je datum obvezen.

3. Organiziran vnos jedilnika

V modul je mogoče vnesti več porcij prehrane:

- zajtrk,
- kosilo,
- večerja,
- prigrizki,
- dodatne opombe.

Ta delitev omogoča jasen in pregleden prikaz za pacienta ter metodično pripravo načrta s strani nutricionista. Podrobni opisi živil, količin ali navodil so možni, ker je vsako polje prilagojeno za vnos besedila.

4. Spreminjanje, dodajanje in odstranjevanje načrtov

Če pacient potrebuje prilagoditve, ima alergije, je dosegel določen napredek ali se njegove prehranske potrebe razlikujejo, lahko nutricionist izvede vse potrebne popravke ali dodatke k prehranskemu načrtu.

Da bi se izognili nenamernemu brisanju, bo sistem pred odstranitvijo testnih ali odvečnih načrtov zahteval potrditev.

Dodatne komponente modula

Profesionalno zelena zasnova

Izbrali smo znano zeleno barvno paleto, ki predstavlja zdravje, uravnoteženo prehrano in naravnost, saj smo želeli dosleden, uglajen in umirjen videz. V pregledni in osnovni zasnovi vmesnika sta prednost berljivost in uporabniška izkušnja.

Nedvoumna postavitev

Tako pacient kot nutricionist bosta informacije zlahka razumela, saj so organizirane v kartice in jasno označene razdelke.

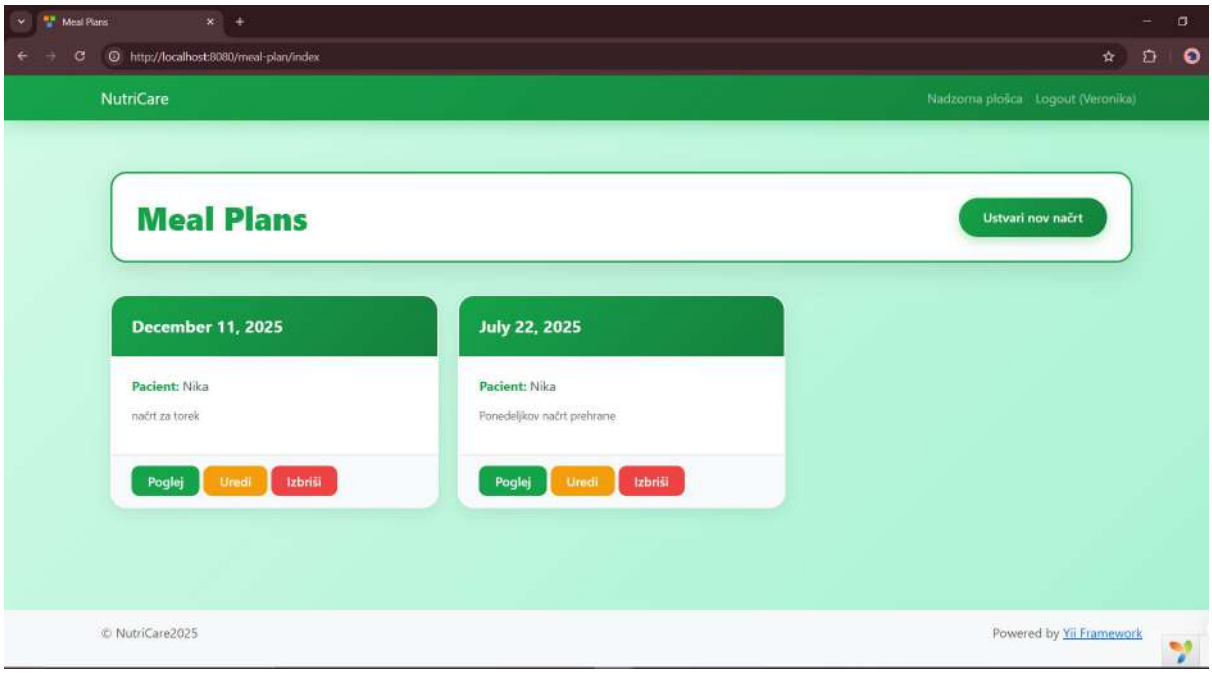
Glavne prednosti postavitve so:

- nedvoumna struktura vnosa,
- vizualna razdelitev različnih obrokov,
- logičen vrstni red podatkov,
- hiter pogled pripravljenih načrtov.

Odsotnost možnosti urejanja ali brisanja s strani pacienta preprečuje nenamerne spremembe in ohranja nadzor nutricionista nad pacientovo prehrano in profesionalnim odnosom. To pomeni, da ima pacient vpogled v enako vsebino, vendar brez gumbov za urejanje in brisanje.

Pomen modula

Pri delu nutricionista nudi modul prehranskega načrta ključno organizacijsko in komunikacijsko podporo, saj združuje vsa prehranska priporočila, omogoča enostavno spremljanje napredka, povečuje motivacijo pacientov ter omogoča digitalno, pregledno in učinkovito sodelovanje pacienta in dietetika.



Slika 7: Prikaz aplikacije NutriCare v ogrodju Yii

Vir: lasten

Sistem klepeta

Ena najbolj zanimivih funkcionalnosti aplikacije NutriCare je modul za klepet. Za daljše prehranske programe ali programe hujšanja je namenjen omogočanju hitrega in stalnega stika med pacientom in nutricionistom.

Ustvarili smo osnovno izkušnjo v realnem času, ki omogoča nemoten dialog z malo prekinitvami.

Zasnova podatkov

Podatkovna struktura, ki se uporablja za urejanje sporočil, vključuje:

- pošiljatelja (sender_id), ki označuje pošiljatelja sporočila;
- prejemnika (receiver_id), ki označuje prejemnika sporočila;
- vsebino sporočila;
- čas pošiljanja (created_at);
- stanje branja (is_read), ki omogoča nadaljnje izboljšave, kot je prikaz »prebrano«.

Takšna struktura omogoča skalabilnost sistema, preprosto filtriranje klepeta in razvrščanje sporočil ter ustvarja osnovo za morebitno analizo.

Značilnosti komunikacijskih sistemov

1. Sporočila – v vnosno polje uporabnik vnese sporočilo. Ob pritisku gumba za pošiljanje se v ozadju na strežnik pošlje tiha zahteva. Podobno kot delujejo pogovori v sodobnih programih, se uporabnik ne zaveda nalaganja ali preusmeritve.

2. Uporaba asinhrone komunikacije (AJAX) – AJAX se uporablja namesto osvežitve le dela strani, kar pomeni, da se izvede le del zahteve in strežnik vrne le potrebne podatke.

Prednosti:

- hitrost,
- manjša obremenitev strežnika,
- izboljšana uporabniška izkušnja,
- občutek komunikacije v realnem času.

3. Samodejno osveževanje sporočil

Sistem vsakih nekaj sekund ugotovi, ali so bila vmes poslana kakšna nova sporočila. Takšen pristop imenujemo periodično ponavljanje (ang. *polling*) in predstavlja enostavnejšo metodo komunikacije v realnem času ter je idealen za manjše aplikacije. Na ta način si lahko uporabnik ogleda nova sporočila, ne da bi moral ponovno naložiti stran.

4. Različne predstavitve glede na pošiljatelja

Sporočila so vizualno ločena za izboljšanje preglednosti in intuitivnosti v pogovoru:

- uporabnikova komunikacija je usmerjena na desno,
- uporabnikova prejeta sporočila so usmerjena na levo.

Slednje daje videz »mehurčkov«, podoben tradicionalnim aplikacijam, kot so Viber, WhatsApp in Messenger.

Označevanje sporočil kot prebranih

Ko prejemnik odpre sporočilo, se status sporočila v podatkovni bazi posodobi na status »prebrano«.

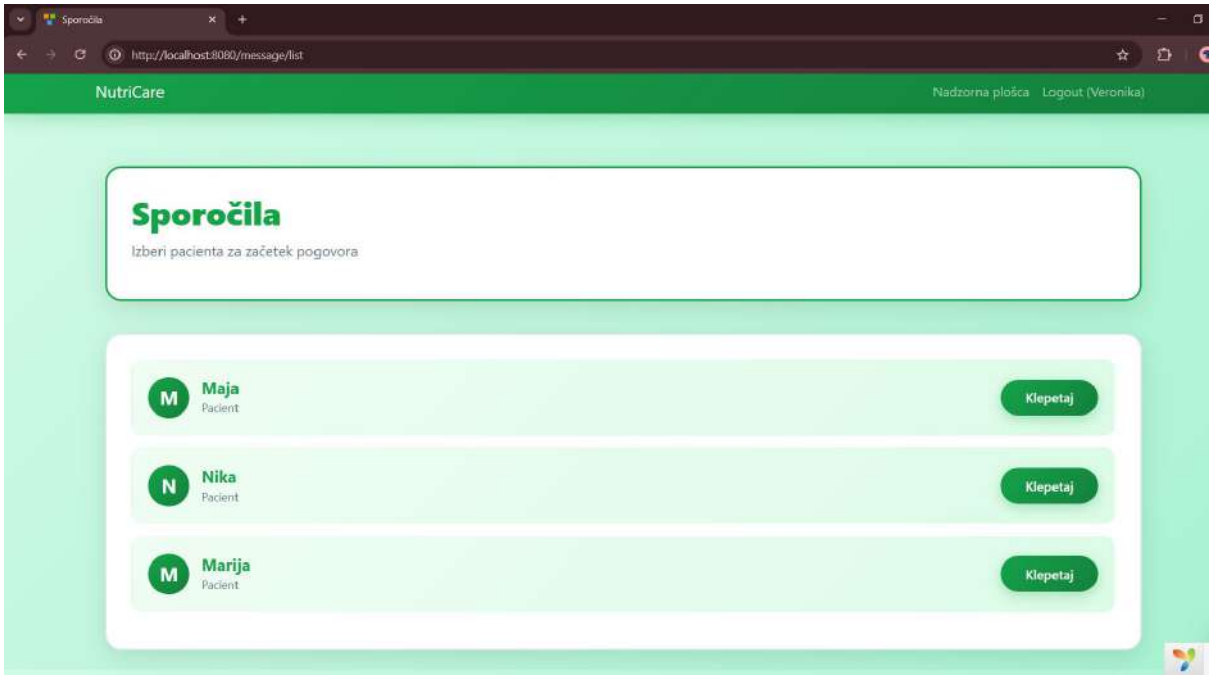
Ta mehanizem omogoča dodatne razširitve, kot so: prikaz prebranih sporočil, izračun števila neprebranih sporočil in opozorila za nova sporočila.

Podrobna razlaga komunikacijskega procesa

- Uporabnik vnese besedilo v vnosno polje;
- JavaScript pošlje strežniku nevidno zahtevo z besedilom, pošiljateljem in prejemnikom;
- po prejemu podatkov strežnik preveri njihovo veljavnost, varnost in dovoljenja, preden jih shrani v podatkovno bazo;
- prednji del vsaki dve sekundi preveri, ali so bila dodana nova sporočila;
- zaznana nova sporočila so prikazana v klepetnem oknu brez ponovnega nalaganja strani;
- sporočila so glede na pošiljatelja takoj organizirana v pogovorne mehurčke;
- uporabnik opazuje nemoten dialog, ki zelo posnema dejanski klepet.

Pomen modula za pobudo NutriCare

Aplikacija za sporočanje omogoča: stalen dialog med dietetikom in pacientom, kratke komentarje o spremembah prehrane, motivacijo pacientov in digitalno pomoč za trajne spremembe življenjskega sloga. Kljub temu da sistem ne uporablja WebSocketov, je dovolj hiter in zanesljiv, da zadosti potrebam manjših prehranskih aplikacij in ga je mogoče v prihodnosti tudi razširiti.



Slika 8: Prikaz aplikacije NutriCare, narejene z ogrodjem Yii

Vir: lasten

4.2.3 Izkušnje z razvojem Yii2

Za izdelavo aplikacije NutriCare je bilo uporabljeno izključno ogrodje Yii2, kar nam je omogočilo odlično in oprijemljivo praktično znanje. Razumevanje arhitekture, orodij in delovnih postopkov Yii2 je bilo ključnega pomena za celoten razvojni proces. Rezultati, prednosti, slabosti in splošna ocena uporabe Yii2 v dejanskem projektu so navedeni spodaj.

Prednosti ogrodja Yii2

1. Gii, zelo zmogljivo orodje za samodejno generiranje kode

Ena največjih prednosti Yii2 je vsekakor Gii, spletni generator kode, ki omogoča, da razvijalec samo izbere tabelo iz baze in Gii samodejno ustvari:

- model,
- krmilnik,
- logiko CRUD (ustvarjanje, branje, urejanje in brisanje),
- poglede/obrazce,
- potrditev,
- osnovne odnose med komponentami.

To pomeni, da lahko uporaba orodja Gii bistveno skrajša čas razvoja osnovnih modulov.

Ta funkcija je bila za nas bistvena, ker je omogočila, da smo se osredotočili na oblikovanje, uporabniško izkušnjo in poslovno logiko aplikacije namesto na ročno implementacijo ponavljajočih funkcionalnosti.

2. ActiveRecord: preproste in pregledne operacije z bazami podatkov

Yii2 uporablja edinstveno tehniko upravljanja baz podatkov, znano kot ActiveRecord.

V resnici to pomeni, da:

- razvijalcu ni treba pisati poizvedb SQL,
- je mogoče z zapisi ravnati kot z objektom,
- je vsak podatkovni zapis povezan na logičen, objektno usmerjen način,
- postopki, ki so preprosti za uporabo, izhajajo iz razmerij med tabelami.

V praksi je to omogočilo:

- hitrejše pridobivanje informacij iz baze podatkov,
- enostavno vzpostavljanje relacij med sporočili, nutricionisti in pacienti,
- objektno usmerjeno upravljanje s podatki, brez ročnega pisanja SQL stavkov.

Poleg tega je to povečalo varnost in zanesljivost, saj ActiveRecord poskrbi za pravilno konstrukcijo poizvedb in se izogne številnim pogostim napakam, ki nastanejo pri ročnem kodiranju SQL, kot so sintaktične težave in vrivanje SQL.

Profesionalnost, doslednost in predvidljivost

Yii2 je zelo zanesljivo ogrodje, ki se uporablja že dolgo časa. Med razvojem smo odkrili naslednje prednosti:

- gradivo je celovito in enostavno za razumevanje,
- dosledna arhitekturna zasnova,
- spremembe različic ne vplivajo na trenutne funkcije,
- projekt ima veliko koristi od počasnega spreminjanja ogrodja,
- primerno je za dolgoročne, profesionalne projekte.

Pri delu smo se naučili, da lahko aplikacija Yii2 deluje več let brez potrebe po dodatnem vzdrževanju, kar je velika prednost za podjetja, ki iščejo stabilno in zanesljivo okolje.

Slabosti ogrodja Yii2

Med razvojem smo odkrili, da ima Yii2 kljub številnim prednostim določene pomanjkljivosti.

1. Manjša razširjenost in skupnosti

   Trenutno velja Laravel za najbolj razširjeno ogrodje PHP.

   Zaradi svoje manjše priljubljenosti ima Yii2: manj programerjev, manj posodobitev, manj vnaprej pripravljenih rešitev na internetu ter manj razširitev in paketov.

   To pomeni, da smo se morali občasno zanašati na dokumentacijo ali pa samostojno najti rešitev.

2. Omejena razpoložljivost učnih virov

   Yii2 je večinoma predstavljen v besedilni dokumentaciji in zastarelih objavah na forumih, medtem ko Laravel vključuje na tisoče sodobnih vadnic, tečajev in videoposnetkov na

YouTubu. Če razvijalec raje sledi interaktivnim smernicam ali video razlagam, se lahko učenje znatno upočasni.

3. Omejena razpoložljivost »začetnih paketov«
   Laravel ponuja najsodobnejše pakete za: preverjanje, API-je, takojšnjo komunikacijo, integracijo v vmesniku in upravljanje uporabnikov.
4. Yii2 je bolj tradicionalen in zahteva več ročne implementacije, kar pomeni: dodaten čas za konfiguracijo, boljše razumevanje arhitekture, manj možnosti, »pripravljenih za uporabo«.
5. Zahtevnejša začetna konfiguracija
   Yii2 uporablja konfiguracijske datoteke (web.php, console.php in db.php).

   Za začetnike je slednje lahko težavno, ker so komponente zelo prilagodljive, zato mora razvijalec razumeti: način delovanja dobaviteljev, način nalaganja razredov ter kako se registrirajo razširitve in moduli. Čeprav je večji nadzor dolgoročna korist, je lahko prvi korak bolj zahteven.

Osebna izkušnja z delom v Yii2

Med ustvarjanjem NutriCare smo odkrili, da Yii2: zagotavlja res trdno osnovo, omogoča hitro gradnjo modulov z uporabo Gii, je zaupanja vreden in varen ter ima organiziran sistem MVC. Kljub temu da ni najbolj »moderno« ogrodje, omogoča profesionalne aplikacije. Največ časa smo posvetili razumevanju ogrodja, zlasti nastavitvi in pravilnemu delu z modelom. Razvoj je postal izjemno učinkovit, ko smo dojeli osnove logike. Menimo, da Yii2 omogoča implementacijo modulov, kot sta »Načrt obrokov« in »Klepet«, na res jasen, razumljiv in dobro organiziran način.

Tabela 5: Primerjava razvoja aplikacij z ogrodjema Laravel in Yii

Kriterij	**Laravel**	**Yii2**
Popularnost	Najbolj razširjeno PHP ogrodje, ogromno posnetkov uporabe, paketov,	Manjša skupnost, manj komercialnih paketov,

	forumi in video vsebine.	stabilna in zanesljiva skupina uporabnikov.
Hitrost razvoja	Zelo hitra – veliko starter paketov, vgrajena avtentikacija, orodja za »fronted«.	Hitro z Gii generatorjem za CRUD in modele, manj »full-stack starter« paketov, več ročnega dela za nekatere funkcije.
Arhitektura in nadzor	Eleganten »developer-friendly« slog z močnimi konvencijami, nekaj več »magičnih« pomočnikov za hitro delo.	Bolj ekspliciten in predvidljiv, manj skritih »magičnih« prečic – več nadzora razvijalcu.
ORM – delo z bazo	Eloquent ORM – zelo prijazen, ekspresivna sintaksa, močna integracija z migracijami in »seederji«.	ActiveReacord – zelo zmogljiv pri kompleksnih relacijah, bolj neposreden (explicit) pristop pri narejenih poizvedbah.
»Real-time« in moderne funkcije	Močne možnosti in ekostitem (Evenets, Queues, Braodcasting, Horizon, Sunctum, Livewire).	Podpira podobne rešitve, vendar moraš pogosto integrirati zunanje pakete in konfigurirati ročno.
Varnost	Dobre varnostne prakse in vgrajena orodja, veliko paketov za avtekntikacijo/avtorizacijo.	Varnost globoko integrirana v jedro, zelo primeren za varnostno občutljive sisteme.
Razširjenost in »enterprise«	Odlična za sodobne arhitekture in mikro-servise, bogat ekosistem paketov.	Zelo primeren za poslovne (enterprise) projekte: stabilnost, deterministična nadgradnja in manj motenj

		pri posodobitvah (breaking changes).
Učna krivulja	Prijazen začetnikom, veliko učnih virov.	Višja začetna krivulja pri konfiguraciji in komponentah, a dobre dolgoročne prednosti.
Kdaj izbrati Laravel/Yii2?	Ko se potrebuje hitro prototipiranje z množico vnaprej pripravljenih modulov, močno »front-end« integracijo in hitro UI implementacijo.	Ko se potrebuje zelo stabilno, varno in visoko kontrolirano rešitev za kompleksne relacije podatkov in dolgoročno vzdrževanje.
Primernost za projekt NutriCare	UI in razvoj hitrosti: Laravel bi pohitril postavitev uporabniških vmesnikov in avtentikacijskih modulov.	Varnost in stabilnost: Yii2 je odličen za NutriCare, ker gre za sistem z občutljivimi podatki, nudi močan nadzor nad varnostjo in stabilnostjo.
Končni nasvet	Primeren tako za hitre prototipe kot za zahtevne produkcijske aplikacije z modernimi UI komponentami.	Če je cilj robustna, varna in dolgoročno vzdrževana produkcijska aplikacija z natančnim nadzorom.

Vir: lastna primerjava na osnovi praktične implementacije

5 MOŽNOSTI ZA NADALJNJI RAZVOJ APLIKACIJE NUTRICARE

Ustvarjena aplikacija NutriCare je varen in uporaben prototip spletne platforme, zasnovane za lažjo komunikacijo med uporabnikom in strokovnjakom za prehrano. Glavne teme diplomskega dela so bile izdelava aplikacij, primerjava ogrodij PHP Laravel in Yii ter uporaba temeljnih varnostnih ukrepov. Kljub temu obstaja več možnosti za nadaljnjo rast in izboljšave obstoječe rešitve, kar bi omogočilo širšo uporabo aplikacije v dejanskem produkcijskem okolju (Sommerville I. , 2016) (Fowler M. , Patterns of Enterprise Application Architecture, 2003).

V nadaljevanju so opisane glavne smernice za prihodnji razvoj aplikacije NutriCare, s poudarkom na povečanju funkcionalnosti, izboljšanju uporabniške izkušnje, krepitvi varnosti in zagotavljanju skladnosti z zakonskimi in zdravstvenimi zahtevami.

Razvoj mobilne aplikacije kot razširitve spletne platforme

Izdelava mobilne aplikacije, ki bi služila kot razširitev trenutne spletne platforme NutriCare, bi bila ena najpomembnejših izboljšav. V sodobnem digitalnem okolju uporabniki vse pogosteje dostopajo do storitev prek pametnih telefonov, zato mobilna podpora pomembno vpliva na dolgoročno uspešnost aplikacije in uporabniško izkušnjo (Sommerville I. , 2016).

Osnovne funkcije, kot so preverjanje načrtov obrokov, pošiljanje sporočil, prejemanje obvestil in sledenje napredku, bi lahko bile hitreje in enostavneje dostopne z mobilno aplikacijo. Posebno prednost bi predstavljala uporaba potisnih obvestil, ki bi uporabnike pravočasno obveščala o novih sporočilih, spremembah prehranskih načrtov ali prihajajočih posvetovanjih (Fowler M. , Patterns of Enterprise Application Architecture, 2003).

Z arhitekturnega vidika lahko mobilna aplikacija izkoristi že obstoječi zaledni sistem Laravel ali Yii z uporabo REST ali GraphQL API-ja. Takšen pristop omogoča ponovno uporabo poslovne logike, zmanjšuje razvojne stroške in zagotavlja enotno varnostno politiko za vse odjemalce (Sommerville I. , 2016). Poleg tega bi se za ustvarjanje mobilne aplikacije lahko uporabile večplatformske tehnologije, kot sta Flutter ali React Native, kar bi dodatno zmanjšalo čas razvoja in stroške vzdrževanja.

Uvedba »end-to-end« šifriranja komunikacije

Čeprav aplikacija NutriCare že uporablja varno komunikacijo prek protokola HTTPS in šifriranje občutljivih podatkov na strežniški strani, bi dodatno stopnjo varnosti predstavljala uvedba »end-to-end« šifriranja komunikacije. »End-to-end« šifriranje zagotavlja, da so podatki šifrirani že na strani pošiljatelja in jih lahko dešifrira izključno prejemnik (Smith, 2016).

Takšna zaščita je še posebej pomembna pri aplikacijah, ki obdelujejo občutljive osebne in zdravstvene podatke, kot so prehranjevalni vzorci, zdravstveni cilji in individualni prehranski načrti. V primeru morebitnega varnostnega incidenta ali nepooblaščenega dostopa do strežnika bi bila vsebina komunikacije še vedno zaščitena (Foundation O. , 2023).

Za implementacijo E2EE so potrebne sodobne kriptografske metode in ustrezno upravljanje kriptografskih ključev na strani odjemalca. Takšna izboljšava bi prispevala k večjem zaupanju uporabnikov, prav tako pa bi program približala varnostnim standardom, ki jih uporabljajo sodobne komunikacijske platforme, kljub dodatni kompleksnosti.

Možnost video svetovanj med uporabniki in prehranskimi strokovnjaki

Uvedba video posvetovanj med uporabniki in nutricionisti bi pomenila pomembno funkcionalno razširitev programa NutriCare. Video komunikacija omogoča bolj oseben pristop k svetovanju, saj vključuje tudi neverbalno komunikacijo, kar lahko izboljša kakovost svetovalnega procesa (WHO, 2021).

Sodobne spletne tehnologije, kot je WebRTC, ki omogoča varen prenos zvoka in videa neposredno v spletnem brskalniku brez potrebe po dodatnih vtičnikih, bi se lahko tehnološko uporabile za izvajanje video posvetovanj (WebRTC, n.d.). Video klici bi delovali skupaj s sistemom za načrtovanje terminov in bi jih bilo mogoče vključiti v trenutni uporabniški vmesnik aplikacije.

Ker se občutljivi podatki izmenjujejo v realnem času, bi bilo z varnostnega vidika nujno zagotoviti šifrirano povezavo, nadzor dostopa in ustrezno upravljanje sej. Takšna posodobitev bo povečala uporabnost NutriCare v profesionalnih zdravstvenih okoljih in ga približala sodobnim možnostim telemedicine.

Integracija z nosljivimi napravami

Integracija z nosljivo tehnologijo, kot so pametne ure in zapestnice, je ključni naslednji korak za razvoj aplikacije NutriCare. Te naprave omogočajo spremljanje telesne aktivnosti, srčnega utripa, porabe kalorij in kakovosti spanja (Google, 2023), (Apple, 2023).

Nutricionisti bi lahko z vključitvijo teh podatkov bolje spremljali življenje uporabnikov in ustvarili bolj individualizirane in uspešne prehranske režime. Poleg tega bi se podatki nosljivih naprav lahko uporabili za prikaz napredka uporabnikov in povečanje njihove motivacije.

S tehničnega vidika bi bila integracija mogoča s trenutnimi API-ji, kot sta Apple HealthKit ali Google Fit. V tem primeru bi bilo ključnega pomena zagotoviti ustrezno obdelavo in zaščito podatkov ter izrecno soglasje uporabnika za zbiranje in uporabo teh podatkov.

Skladnost z GDPR in dodatnimi predpisi o zdravju in varnosti

V primeru produkcijske uporabe aplikacije NutriCare bi bilo ključnega pomena strogo upoštevanje zakonskih in regulativnih predpisov, zlasti Splošne uredbe o varstvu podatkov (GDPR) (European Union, 2016). To vključuje pregledno ravnanje z osebnimi podatki, dobro opredeljene politike zasebnosti in zaščito uporabniških pravic, vključno z možnostjo dostopa do podatkov, njihovega posodabljanja in odstranjevanja. Poleg GDPR bi bilo treba upoštevati tudi druge varnostne in zdravstvene standarde, kot je ISO/IEC 27001, ki vzpostavlja sistem upravljanja informacijske varnosti. Zanesljivost sistema bi se še povečala z dodatnimi varnostnimi funkcijami, kot so večfaktorska avtentikacija, revizijske sledi, sledenje dostopa in rutinski varnostni pregledi.

Pregled prihodnjega razvoja

Aplikacija NutriCare ima v skladu s predlaganimi možnostmi prihodnjega razvoja veliko potenciala za izboljšanje v celovito, varno in sodobno platformo za digitalno prehransko svetovanje. Aplikacija bi postala konkurenčna rešitev za e-zdravje z razširitvijo na mobilne naprave, dodajanjem naprednih varnostnih funkcij, omogočanjem video posvetovanj in integracijo z nosljivo tehnologijo. Nadaljnji razvoj bi omogočil večjo uporabniško vrednost, večjo varnost ter upoštevanje zakonskih in strokovnih zahtev.

6 SKLEP

Doseganje ciljev diplomskega dela

V diplomskem delu je bil obravnavan razvoj spletne aplikacije NutriCare, katere namen je varna obdelava osebnih in prehranskih podatkov ter olajšana komunikacija med uporabniki in strokovnjaki za prehrano. Glavni cilji diplomskega dela so bili razviti delujočo spletno aplikacijo, analizirati varnostne vidike, ki so ključni pri razvoju aplikacij na področju prehrane in zdravja, ter primerjati dve uveljavljeni PHP ogrodji, Laravel in Yii. Poleg tehničnega vidika je bil poudarek tudi na razumevanju pomena varnosti in varstva podatkov v sodobnih spletnih rešitvah.

Cilji diplomskega dela so bili v celoti doseženi. V teoretičnem delu so bila predstavljena ogrodja PHP, pomen skupnosti razvijalcev ter osnovni in naprednejši varnostni mehanizmi spletnih aplikacij. Posebna pozornost je bila namenjena varnostnim funkcionalnostim ogrodij Laravel in Yii, kot so upravljanje sej, šifriranje podatkov, preverjanje pristnosti uporabnikov ter zaščita pred najpogostejšimi spletnimi napadi, med katerimi so vrivanje SQL, XSS in CSRF.

Spletna aplikacija NutriCare je bila v praktičnem delu razvita kot študija primera, s katero je bilo mogoče preveriti uporabnost in varnost obeh ogrodij v realnem razvojnem okolju. Razvoj aplikacije je omogočil neposredno primerjavo ogrodij ter pridobitev praktičnih izkušenj, ki so prispevale k bolj objektivni in uravnoteženi oceni njunih prednosti in omejitev. Povezovanje teoretičnih spoznanj s praktično implementacijo je bilo uspešno doseženo.

Hipoteza H3, da obe ogrodji ob pravilni in strokovni uporabi zagotavljata primerljivo raven varnosti, je bila potrjena. Tako Laravel kot Yii ponujata obsežen nabor vgrajenih in preverjenih varnostnih mehanizmov, vendar se razlikujeta v načinu njihove uporabe. Laravel razvijalcem ponuja bolj avtomatizirane in prijazne rešitve, ki zmanjšujejo verjetnost napačne implementacije varnostnih mehanizmov. Yii pa omogoča bolj ekspliciten in tehnično natančen nadzor nad konfiguracijo in varnostnimi nastavitvami, kar lahko izboljša preglednost in olajša revizijo kode. To ne pomeni, da je Laravel manj varen, temveč da Yii zaradi manjše stopnje t. i. »magije« razvijalcu omogoča bolj neposreden vpogled v delovanje varnostnih mehanizmov.

Prav tako je bila potrjena teza, da uporaba sodobnega ogrodja PHP pomembno izboljša varnost aplikacije v primerjavi z ročno implementacijo varnostnih rešitev v čistem PHP. Razvoj aplikacije NutriCare je pokazal, da Laravel s svojim obsežnim ekosistemom knjižnic, paketov in orodij omogoča hitrejši razvoj aplikacij, ne da bi se pri tem zmanjšala raven varnosti.

Ključne ugotovitve raziskave

Ena ključnih ugotovitev diplomskega dela je, da Laravel in Yii predstavljata zmogljivi in varni PHP ogrodji, ki sta primerni za različne razvojne pristope in vrste projektov. Laravel izstopa po hitrosti razvoja, sodobnih razvojnih praksah, dobri uporabniški izkušnji za razvijalce ter močni podpori skupnosti. Zaradi teh lastnosti je posebej primeren za projekte, kjer sta hitra implementacija in krajši čas do zagona aplikacije ključnega pomena.

Yii se je izkazal kot zelo stabilno in tehnično dosledno ogrodje, ki poudarja jasno arhitekturo MVC, natančno konfiguracijo in večjo eksplicitnost pri implementaciji funkcionalnosti. Takšen pristop lahko predstavlja prednost pri dolgoročno vzdrževanih sistemih in v okoljih, kjer sta preglednost kode ter enostavnejša varnostna revizija še posebej pomembni. To pa ne pomeni, da je Yii inherentno bolj varen, temveč da njegov razvojni slog omogoča več neposrednega nadzora nad varnostnimi mehanizmi.

Raziskava je pokazala, da ne obstaja univerzalno idealno ogrodje PHP, saj ima izbira ogrodja pomemben vpliv na kakovost, vzdržljivost, varnost in dolgoročni uspeh spletne aplikacije, hkrati pa je močno odvisna od zahtev projekta in izkušenj razvojne ekipe.

Zaključne opombe

Diplomsko delo je potrdilo, da ne obstaja eno samo ogrodje PHP, ki bi bilo optimalno za vse vrste projektov, saj je odločitev vedno pogojena s specifičnimi zahtevami, obsegom aplikacije in pričakovano rastjo sistema. Izbira ustreznega ogrodja lahko pomembno prispeva k večji varnosti, boljši uporabniški izkušnji in večji vzdržljivosti aplikacije, kar je bilo potrjeno tudi s študijo primera aplikacije NutriCare.

Zaključno delo predstavlja dobro izhodišče za nadaljnje raziskave, kot so primerjave z drugimi ogrodji PHP, analiza zmogljivosti in skalabilnosti v realnih okoljih ali poglobljeno varnostno testiranje. S tem diplomsko delo ne zaključuje obravnavane teme, temveč odpira možnosti za njen nadaljnji razvoj.

VIRI IN LITERATURA

Apple. (2023). *HealthKit framework overview*. Pridobljeno iz https://developer.apple.com/documentation/healthkit

Bootstrap. (2024). *Bootstrap*. Pridobljeno iz Bootstrap 5 documentation: https://getbootstrap.com/docs/5.3/

Bučar, F. (2012). *Rojstvo države.* Ljubljana: Didakta.

Community, L. (2025). *Laravel Community*. Pridobljeno iz Laravel Community: https://laravel.com/community

Composer. (2024). *Composer*. Pridobljeno iz Composer documentation: https://getcomposer.org/doc/

European Union. (2016). *General Data Protection Regulation.*

Foundation, O. (2023). *OWASP Foundation*. Pridobljeno iz OWASP Top 10 – Web application security risks: https://owasp.org/www-project-top-ten/

Fowler, M. (2003). *Patterns of Enterprise Application Architecture.*

Framework, Y. (2024). *Yii Framework*. Pridobljeno iz Yii 2.0 official documentation: https://www.yiiframework.com/doc/guide/2.0/en

GitHub. (2026). *Laravel and Yii repositories statistics*. Pridobljeno iz GitHub: https://github.com

Google. (2023). *Google Fit API documentation*. Pridobljeno iz https://developers.google.com/fit

Group, P. (2024). *PHP Group*. Pridobljeno iz PHP manual – Security: https://www.php.net/manual/en/security.php

International Organization for Standardization. Pridobljeno iz https://www.iso.org/home.html

Laravel. (2025). Pridobljeno iz Laravel documentation: https://laravel.com/docs

NIST. (2012). *National Institute of Standards and Technology.*

Otwell, T. (2024). *Laravel framework.* Pridobljeno iz https://github.com/laravel/framework

Overflow, S. (2026). *Stack Overflow*. Pridobljeno iz Developer survey and tag statistics: . https://stackoverflow.com/tags

Smith, G. &. (2016). *The Cryptopals Crypto Challenges*. Pridobljeno iz The Cryptopals Crypto Challenges: https://www.cryptopals.com/

Sommerville. (2016). Software engineering. 812.

Stuttard.D and Pinto, M. (2011). *The Web Application Hacker's Handbook.*

Team, B. (2024). *Bootstrap Team*. Pridobljeno iz Bootstrap 5 Documentation: https://getbootstrap.com/

WebRTC. (brez datuma). *WebRTC Project*. Pridobljeno iz WebRTC overview: https://webrtc.org/?utm_source=chatgpt.com

WHO. (2021). *World Health Organization*. Pridobljeno iz Global strategy on digital health 2020–2025: https://www.who.int/%E2%80%A6

Yii Documentation. (brez datuma). Pridobljeno iz https://www.yiiframework.com/doc/guide/2.0/en

Adevait. (2025). Laravel Developer Survey 2025. Pridobljeno iz https://adevait.com/laravel/laravel-developer-survey-2025