UNIVERSITETI “ISA BOLETINI” MITROVICË

FAKULTETI I INXHINIERISË MEKANIKE DHE KOMPJUTERIKE

DEPARTAMENTI: SHKENCA KOMPJUTERIKE DHE INXHINIERI

PUNIM DIPLOME

Mentor:

Agon Bajgora, PhD(c)

Kandidati:

Shefkie Segashi

Mitrovicë, 2026

UNIVERSITETI “ISA BOLETINI” MITROVICË

FAKULTETI I INXHINIERISË MEKANIKE DHE KOMPJUTERIKE

DEPARTAMENTI: SHKENCA KOMPJUTERIKE DHE INXHINIERI

PUNIM DIPLOME

Tema:

Dizajnimi dhe Implementimi i një Platforme Adaptive për Mësim të Vetëdrejtuar të Bazuar në Dëshmi

Mentor: **Kandidati:**

Agon Bajgora, PhD(c) Shefkie Segashi

Mitrovicë, 2026

UNIVERSITY “ISA BOLETINI” MITROVICA

FACULTY OF MECHANICAL AND COMPUTER ENGINEERING

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING

BACHELOR THESIS

Topic Title:

Design and Implementation of an Adaptive Proof-Based Platform for Self-Directed Learning

Mentor: **Candidate:**

Agon Bajgora, PhD(c) Shefkie Segashi

Mitrovicë, 2026

DEKLARATË E ORIGJINALITETIT

Unë, Shefkie Segashi, deklaroj se ky punim diplome me titull *"Dizajnimi dhe Implementimi i një Platforme Adaptive për Mësim të Vetëdrejtuar të Bazuar në Dëshmi"* është punë origjinale e imja, e punuar nën udhëheqjen e mentorit tim. Të gjitha burimet e përdorura janë cituar sipas rregullave akademike dhe janë listuar në seksionin e literaturës. Ky punim nuk është dorëzuar më parë, tërësisht ose pjesërisht, për marrjen e ndonjë grade tjetër akademike.

Aplikacioni *QuestPath* i përshkruar në këtë punim është zhvilluar si prototip funksional. Përshkrimet e implementimit dhe testimit paraqesin gjendjen reale të aplikacionit gjatë kohës së hartimit të punimit.

Kandidati: Shefkie Segashi Nënshkrimi: _____ Data: _____

FALËNDERIME

Së pari, dëshiroj të shpreh mirënjohjen dhe falënderimin tim më të sinqertë për familjen time, për mbështetjen, motivimin dhe përkrahjen e pakushtëzuar gjatë gjithë rrugëtimit tim akademik. Mbështetja dhe besimi i tyre kanë qenë një shtysë e rëndësishme për mua në arritjen e këtij qëllimi.

Një falënderim të veçantë ia drejtoj mentorit tim, për ndihmën, udhëzimet dhe këshillat e vlefshme gjatë realizimit të këtij punimi. E falënderoj gjithashtu për besimin, përkrahjen dhe gatishmërinë për të më ndihmuar gjatë gjithë procesit të punimit të diplomës.

Në fund, falënderoj të gjithë ata që, në çfarëdo mënyre, më kanë mbështetur dhe kanë kontribuar në përfundimin e këtij punimi.

ABSTRAKT

Mësimi i vetëdrejtuar po bëhet gjithnjë e më i rëndësishëm, veçanërisht në fushat teknike, ku studentët shpesh duhet të zhvillojnë njohuri dhe aftësi edhe jashtë mësimit tradicional. Megjithatë, shumë platforma ekzistuese e paraqesin përparimin kryesisht përmes pikëve, kuizeve ose përfundimit të detyrave, pa kërkuar gjithmonë dëshmi konkrete për punën e realizuar.

Në këtë punim është dizajnuar dhe zhvilluar *QuestPath*, një platformë adaptive për mësim të vetëdrejtuar të bazuar në dëshmi. Platforma i mundëson përdoruesit të përcaktojë një qëllim që dëshiron ta arrijë dhe, në bazë të tij, të ndjekë një rrugë mësimi të ndarë në misione të vogla ditore. Për secilin mision, përdoruesi paraqet dëshmi për punën e kryer dhe jep një reflektim mbi atë që ka mësuar. Në bazë të këtyre të dhënave, sistemi vlerëson progresin dhe përshtat hapat e ardhshëm të rrugëtimit të tij.

Platforma është zhvilluar si aplikacion web duke përdorur Next.js, TypeScript, PostgreSQL dhe Prisma ORM. Gjithashtu, është implementuar autentikimi i përdoruesve dhe ruajtja e sigurt e fjalëkalimeve. Sistemi përfshin funksionalitete për krijimin e misioneve, dorëzimin e dëshmive, reflektimin e përdoruesit, përshtatjen e misioneve të ardhshme dhe mundësinë që mentori të shqyrtojë punën e dorëzuar dhe të japë komente.

Funksionalitetet kryesore të platformës janë testuar gjatë zhvillimit për të kontrolluar funksionimin e tyre, paraqitjen në pajisje të ndryshme dhe funksionimin e aplikacionit pas publikimit. Në këtë fazë, vlerësimi është bërë mbi funksionimin e prototipit dhe skenarë testues dhe nuk është realizuar ende një studim me përdorues realë. Mekanizmi adaptiv aktual bazohet në rregulla të përcaktuara në sistem dhe jo në metoda të inteligjencës artificiale apo të mësimit makinerik. Në fund, paraqiten kufizimet aktuale të platformës dhe mundësitë për zhvillimin dhe përmirësimin e saj në të ardhmen.

Fjalë kyçe: mësim i vetëdrejtuar, mësim adaptiv, vlerësim i bazuar në dëshmi, reagim formativ, platformë adaptive, aplikacion web.

ABSTRACT

Self-directed learning is becoming increasingly important, particularly in technical fields, where students are often expected to develop knowledge and skills beyond traditional classroom learning. However, many existing platforms represent progress mainly through points, quizzes, or task completion, without always requiring concrete evidence of the work performed.

This thesis presents the design and development of *QuestPath*, an adaptive platform for evidence-based self-directed learning. The platform allows users to define a goal they want to achieve and, based on that goal, follow a learning path divided into small daily missions. For each mission, users submit evidence of the work they have completed and provide a reflection on what they have learned. Based on this information, the system evaluates their progress and adapts the next steps in their learning path.

The platform was developed as a web application using Next.js, TypeScript, PostgreSQL, and Prisma ORM. User authentication and secure password storage have also been implemented. The system includes functionalities for creating missions, submitting evidence, recording user reflections, adapting future missions, and allowing mentors to review submitted work and provide feedback.

The main functionalities of the platform were tested during development to verify their operation, their presentation across different devices, and the application's functionality after deployment. At this stage, the evaluation is based on the functionality of the prototype and test scenarios, and no study involving real users has yet been conducted. The current adaptive mechanism is based on predefined system rules rather than artificial intelligence or machine learning methods. Finally, the thesis presents the current limitations of the platform and discusses possibilities for its future development and improvement.

Keywords: self-directed learning, adaptive learning, evidence-based assessment, formative feedback, adaptive platform, web application.

TABELA E PËRMBAJTJES

LISTA E FIGURAVE

LISTA E TABELAVE

1 HYRJE

1.1 Konteksti dhe motivimi

Mënyra se si njerëzit mësojnë aftësi teknike ka ndryshuar thellësisht gjatë dekadës së fundit. Një pjesë gjithnjë e më e madhe e të mësuarit ndodh jashtë sallës tradicionale të mësimit: përmes dokumentacionit online, video-mësimeve, projekteve personale dhe komuniteteve profesionale. Ky model i *mësimit të vetëdrejtuar* i jep nxënësit liri të madhe, por njëkohësisht e ngarkon atë me përgjegjësinë e plotë për planifikimin, ritmin, vlerësimin dhe ruajtjen e motivimit (Knowles, 1975). Pikërisht këtu shfaqet hendeku: liria pa strukturë shpesh kthehet në humbje vrulli, ndërsa struktura pa dëshmi reale kthehet në një ndjesi rrejshme përparimi.

Shumica e mjeteve dixhitale që pretendojnë se ndihmojnë mësimin e vetëdrejtuar e matin progresin përmes përfundimit të leksioneve, përgjigjeve të kuizeve ose grumbullimit të pikëve. Këto sinjale janë të lehta për t'u matur, por të dobëta si dëshmi: një nxënës mund të "përfundojë" një kurs pa prodhuar asnjë artefakt që dikush tjetër mund ta inspektojë. Kjo shkëputje midis *ndjesisë së progresit* dhe *dëshmisë së progresit* është problemi qendror që adreson ky punim. Kur progresi matet me pikë të shkëputura nga puna reale, motivimi bëhet i brishtë dhe i manipulueshëm, dhe aftësia e fituar mbetet e paverifikueshme.

QuestPath lind nga një bindje e thjeshtë: përparimi duhet të jetë i *fituar* dhe i *dëshmuar*, jo i pretenduar. Në vend që ta pyesë nxënësin nëse e ka "kuptuar" një temë, platforma i kërkon të prodhojë një dëshmi konkrete — një lidhje drejt punës, një pamje ekrani, një artefakt, një reflektim të shkruar — dhe pikërisht kjo dëshmi bëhet njësia bazë e progresit. Mbi këtë dëshmi ndërtohet edhe adaptimi: rruga e mëtejshme ndryshon në varësi të cilësisë dhe besimit që shoqërojnë dëshminë e dorëzuar.

1.2 Përkufizimi i problemit

Problemi që trajton ky punim mund të formulohet në tri nënprobleme të ndërlidhura:

1. **Progres i pavërtetuar.** Mjetet ekzistuese e barazojnë "aktivitetin" (shikim video, klikim, kuiz) me "të mësuarit", duke prodhuar metrika progresi që nuk korrespondojnë me aftësi të demonstrueshme.

2. **Mungesa e përshtatjes sipas progresit të përdoruesit.** Edhe kur platformat ofrojnë "personalizim", ai zakonisht bazohet në përgjigje të sakta/të gabuara në kuize, jo në vlerësimin e një produkti real të punës.
3. **Reagim i dobët dhe i vonuar.** Reagimi formativ — që literatura e identifikon si një nga faktorët më të fuqishëm të mësimnxënies (Black & Wiliam, 1998; Hattie & Timperley, 2007) — shpesh mungon plotësisht në mjedise vetëmësimi, ose është gjenerik dhe i pavlefshëm.

Si rrjedhojë, nxënësi i vetëdrejtuar mbetet pa tre gjëra thelbësore: një *hap të qartë pasues* të bazuar në atë që sapo bëri, një *dëshmi të verifikueshme* të asaj që arriti, dhe një *cikël reagimi* që e bën përparimin të ndihet i merituar. QuestPath synon t'i adresojë të tria njëkohësisht.

1.3 Qëllimi dhe objektivat

Qëllimi i përgjithshëm i këtij punimi është dizajnimi dhe implementimi i një prototipi funksional të një platforme adaptive për mësim të vetëdrejtuar të bazuar në dëshmi. Platforma synon të mbështesë përdoruesin në arritjen e qëllimeve të tij përmes misioneve të strukturuara, dorëzimit të dëshmive për punën e realizuar dhe përshtatjes së hapave të ardhshëm sipas progresit të tij

Objektivat specifike janë:

- **O1.** Të analizohen parimet kryesore të mësimit të vetëdrejtuar, vlerësimit të bazuar në dëshmi dhe reagimit formativ, si bazë për përcaktimin e kërkesave të sistemit.
- **O2.** Të projektohet arkitektura e aplikacionit web, e cila mundëson menaxhimin e përdoruesve, misioneve, dëshmive, reagimeve dhe procesit të përshtatjes.
- **O3.** Të implementohet një model i të dhënave që përfshin përdoruesit, rrugët e të mësuarit, misionet, dëshmitë e dorëzuara, reagimet, aftësitë dhe arritjet.
- **O4.** Të implementohet mekanizmi për gjenerimin e misioneve dhe përshtatjen e hapave të ardhshëm në bazë të informacionit të dhënë nga përdoruesi gjatë realizimit të tyre.
- **O5.** Të testohet dhe verifikohet funksionimi i prototipit, duke përfshirë funksionalitetet kryesore, paraqitjen në pajisje të ndryshme dhe funksionimin e aplikacionit pas publikimit.
- **O6.** Të identifikohen kufizimet aktuale të prototipit dhe të paraqiten mundësitë për zhvillimin dhe përmirësimin e tij në të ardhmen, duke përfshirë edhe mundësinë e vlerësimit me përdorues realë.

1.4 Pyetjet kërkimore

Punimi udhëhiqet nga pyetjet e mëposhtme:

- **PK1.** Si mund të përdoret dëshmia e punës së realizuar si element kryesor për përcjelljen e progresit në një platformë të mësimit të vetëdrejtuar?
- **PK2.** Cilat të dhëna të ofruara nga përdoruesi gjatë përfundimit të një misioni mund të përdoren për përshtatjen e hapave të ardhshëm, pa përdorur metoda të mësimit makinerik?
- **PK3.** Si mund të projektohet arkitektura e sistemit për të mbështetur procesin e dorëzimit të dëshmisë, reagimit dhe përshtatjes së misioneve, duke ruajtur një strukturë të qartë dhe të mirëmbajtshme?
- **PK4.** Si mund të verifikohet me ndershmëri cilësia e një prototipi të tillë në mungesë të një studimi formal me përdorues?

1.5 Kontributet e punimit

Kontributet kryesore të këtij punimi janë:

1. **Një model konceptual i progresit të bazuar në dëshmi**, ku progresi i përdoruesit lidhet me dëshmitë konkrete të punës së realizuar dhe jo vetëm me pikët ose përfundimin e aktiviteteve.
2. **Një prototip funksional i vendosur në produksion** i platformës QuestPath, me kod të strukturuar dhe të dokumentuar.
3. **Një mekanizëm adaptiv i bazuar në rregulla**, i cili përshtat misionet e ardhshme duke përdorur të dhënat e ofruara nga përdoruesi gjatë realizimit të misioneve, pa përdorur metoda të mësimit makinerik.
4. **Testimi dhe verifikimi i prototipit përmes skenarëve të përcaktuar**, duke dokumentuar funksionimin e tij dhe kufizimet e identifikuara gjatë procesit të zhvillimit.
5. **Një dizajn i propozuar vlerësimi empirik** për punën e ardhshme, që e bën platformën të testueshme shkencërisht.

1.6 Metodologjia në përmbledhje

Punimi ndjek paradigmën e *Shkencës së Dizajnit* (Design Science Research), në të cilën njohuria prodhohet përmes ndërtimit dhe vlerësimit të një artefakti — këtu, prototipit QuestPath. Procesi përfshiu: rishikimin e literaturës për të nxjerrë parimet; nxjerrjen e

kërkesave funksionale dhe jofunksionale; projektimin e arkitekturës dhe modelit të të dhënave; implementimin iterativ; dhe verifikimin përmes kontrolleve teknike dhe validimit me skenarë.

1.7 Fushëveprimi dhe kufizimet

Ky punim përqendrohet te *dizajni dhe implementimi* i një prototipi funksional, jo te vlerësimi empirik i tij me përdorues realë. Si rrjedhojë:

- Nuk është kryer asnjë studim me përdorues, eksperiment i kontrolluar apo anketë; çdo pohim mbi efektivitetin pedagogjik mbetet hipotezë e mbështetur nga literatura, jo gjetje empirike e këtij punimi.
- Motori adaptiv është *heuristik dhe i bazuar në rregulla*, jo një model i mësimit makinerik; ky është një vendim i qëllimshëm dizajni në favor të transparencës.
- Ndërfaqja e mentorit në prototip funksionon me të dhëna shembull të paracaktuara, pasi implementimi i plotë i funksionaliteteve të mentorit nuk është pjesë e kësaj faze.
- Reagimi automatik (“mentor i simuluar”) gjenerohet nga rregulla deterministike; ai shërben si zëvendësues funksional derisa rishikimi njerëzor të integrohet plotësisht me bazën e të dhënave.

Këto kufizime nuk janë dobësi të fshehura, por kufij të deklaruar të fushëveprimit.

1.8 Struktura e punimit

Punimi është organizuar në dymbëdhjetë kapituj. Pas kësaj Hyrjeje, *Kapitulli 2* paraqet literaturën dhe sfondin teorik. *Kapitulli 3* përshkruan metodologjinë. Kapitulli 4 shtjellon kërkesat dhe analizën e sistemit. *Kapitulli 5* paraqet arkitekturën, *Kapitulli 6* implementimin, dhe *Kapitulli 7* modelin e të dhënave. Kapitulli 8 trajton logjikën adaptive të bazuar në dëshmi. *Kapitulli 9* dokumenton testimin dhe verifikimin, *Kapitulli 10* sigurinë dhe privatësinë, *Kapitulli 11* kufizimet dhe punën e ardhshme, dhe *Kapitulli 12* jep përfundimin. Në fund vijnë literatura dhe shtojcat.

2 LITERATURA DHE SFONDI TEORIK

Ky kapitull ndërton bazën teorike mbi të cilën qëndron QuestPath. Qëllimi nuk është një rishikim shterues i të gjithë fushës së teknologjisë arsimore, por nxjerrja e parimeve të qarta, të mbështetura në literaturë, që e udhëheqin dizajnin. Në fund të kapitullit këto parime përmblidhen dhe lidhen drejtpërdrejt me vendimet e dizajnit.

2.1 Mësimi i vetëdrejtuar

Koncepti i mësimit të vetëdrejtuar u formalizua nga Knowles (1975), i cili e përshkroi si një proces ku individët marrin nismën — me ose pa ndihmën e të tjerëve — për të diagnostikuar nevojat e tyre të të mësuarit, për të formuluar synime, për të identifikuar burime, për të zgjedhur strategji dhe për të vlerësuar rezultatet. Kjo qasje është veçanërisht e përshtatshme për të rriturit dhe për fushat teknike që ndryshojnë shpejt, ku njohuria e fituar në një kurs formal vjetërohet me shpejtësi.

Sfida qendrore e SDL-së është vetërregullimi. Nxënësi duhet të menaxhojë njëkohësisht *çfarë* të mësojë, *si* të mësojë dhe *si ta dijë* se ka mësuar. Hulumtimet mbi vetërregullimin tregojnë se nxënësit shpesh e mbivlerësojnë progresin e tyre kur mungojnë sinjale të jashtme objektive. Pikërisht këtu dëshmia konkrete luan një rol të dyfishtë: ajo është njëkohësisht produkt i të mësuarit dhe mjet i vetëvlerësimit më të saktë. Kur nxënësi detyrohet të prodhojë një artefakt që dikush tjetër mund ta inspektojë, hendeku midis "ndjenjës nuk e di" dhe "aftësisë për të bërë" bëhet i dukshëm.

2.2 Mësimi për zotërim dhe problemi 2-sigma

Bloom (1968) propozoi *mësimin për zotërim* (mastery learning), ku nxënësit nuk kalojnë në njësinë e radhës derisa të demonstrojnë zotërim të njësisë aktuale. Më vonë, Bloom (1984) formuloi të famshmin *problem 2-sigma*: nxënësit që mësojnë me udhëzim individual (tutoring) një-me-një performojnë rreth dy devijime standarde më mirë se ata në mësim grupor tradicional. Sfida që ai parashtroi ishte: si të riprodhohet efekti i tutorit individual me metoda të shkallëzueshme?

QuestPath nuk pretendon ta zgjidhë problemin 2-sigma, por dizajni i tij frymëzohet drejtpërdrejt nga ai. Cikli "mision i vogël → dëshmi → reagim → adaptim" është një përpjekje për ta përafruar përvojën e nxënësit me dinamikën e një tutori: vlerësim i shpeshtë, reagim i

menjëhershëm dhe rregullim i hapit pasues sipas performancës. Ndarja e rrugës në misione të vogla ditore pasqyron parimin e zotërimit njësi-pas-njësie.

2.3 Reagimi formativ

Black dhe Wiliam (1998) treguan, përmes një rishikimi të gjerë, se vlerësimi formativ - vlerësimi që përdoret për të informuar të mësuarit e mëtejshëm, jo vetëm për të dhënë notë ka një nga efektet më të mëdha të dokumentuara mbi arritjet. Hattie dhe Timperley (2007) e thelluan këtë me modelin e tyre të fuqishëm të reagimit, i cili përgjigjet në tri pyetje: *Ku po shkoj?* (synimet), *Si po shkoj?* (progresi aktual), dhe *Ku të shkoj më pas?* (hapi pasues). Ata theksojnë se reagimi më efektiv operon në nivelin e procesit dhe të vetërregullimit, jo thjesht në nivelin e detyrës apo të personit.

Ky model ka ndikuar drejtpërdrejt strukturën e reagimit në QuestPath. Çdo reagim i gjeneruar përmban fusha të dedikuara që pasqyrojnë pyetjet e Hattie-t: çfarë shkoi mirë, çfarë mungon, si të përmirësohet dhe cili është veprimi pasues. Vendimi adaptiv përfaqëson eksplicitisht pyetjen “ku të shkoj më pas”.

2.4 Praktika e qëllimshme dhe vetëefikasiteti

Ericsson, Krampe dhe Tesch-Römer (1993) treguan se ekspertiza ndërtohet jo thjesht nga përvoja, por nga *praktika e qëllimshme* (deliberate practice): aktivitete të strukturuara, me vështirësi të përshtatur, me reagim të menjëhershëm dhe me përsëritje të fokusuar te dobësitë. QuestPath e mishëron këtë parim duke i kërkuar nxënësit, pas çdo dorëzimi, të identifikojë pjesën më të dobët dhe duke e adaptuar misionin pasues që ta forcojë pikërisht atë.

Bandura (1977) prezantoi konceptin e *vetëefikasitetit* — besimi i një individi në aftësinë e vet për të kryer një detyrë. Vetëefikasiteti ushqehet kryesisht nga *përvojat e zotërimit*: sukseset konkrete dhe të dëshmuara. Duke e bërë çdo arritje të vogël të prekshme dhe të verifikueshme, QuestPath synon të ndërtojë vetëefikasitet të bazuar në dëshmi reale, jo në inkurajim bosh. Vetëraportimi i besimit gjatë dorëzimit është gjithashtu një masë e drejtpërdrejtë e vetëefikasitetit të perceptuar, e cila përdoret si sinjal adaptiv.

2.5 Zona e zhvillimit proksimal dhe ngarkesa kognitive

Vygotsky (1978) përshkroi *zonën e zhvillimit proksimal* (ZPD): hapësirën midis asaj që nxënësi mund të bëjë vetëm dhe asaj që mund të bëjë me udhëzim. Mësimi efektiv ndodh kur detyrat janë brenda kësaj zone — as shumë të lehta, as të pamundura. Motori adaptiv i QuestPath

përpiqet ta mbajë nxënësin brenda ZPD-së: kur besimi dhe cilësia janë të larta, niveli ngrihet; kur janë të ulëta, misioni përsëritet me kërkesa më të vogla.

Sweller (1988) zhvilloi *teorinë e ngarkesës kognitive*, sipas së cilës kapaciteti i kujtesës së punës është i kufizuar dhe dizajni i mësimit duhet të minimizojë ngarkesën e panevojshme. Kjo justifikon vendimin e QuestPath për t'i mbajtur misionet të vogla dhe të fokusuara në një veprim të vetëm prodhues dëshmie, në vend të detyrave të mëdha e të mjegullta. Ndërfaqja gjithashtu ndjek këtë parim: çdo ekran e bën të qartë veprimin e radhës, duke reduktuar ngarkesën e vendimmarrjes.

2.6 Motivimi: nga gamifikimi sipërfaqësor te dëshmia

Deterding et al. (2011) e përkufizuan *gamifikimin* si përdorimin e elementeve të dizajnit të lojërave në kontekste që nuk janë lojëra. Megjithatë, kritika ndaj gamifikimit sipërfaqësor — pikë, shenja dhe tabela renditjeje të shkëputura nga vlera reale — është e gjerë: kur shpërblimet e jashtme zëvendësojnë qëllimin e brendshëm, ato mund ta dëmtojnë motivimin afatgjatë. Ryan dhe Deci (2000), përmes *teorisë së vetëvendosjes* (self-determination theory), identifikojnë tri nevoja psikologjike themelore: autonominë, kompetencën dhe lidhshmërinë. Motivimi i qëndrueshëm vjen kur këto nevoja plotësohen, jo kur grumbullohen pikë boshe.

QuestPath e pranon vlerën e strukturës së ngjashme me lojën (misione, nivele, XP), por e ankoron çdo element te dëshmia reale: XP fitohet vetëm pas dorëzimit të dëshmisë, dhe progresi është gjithmonë i lidhur me një artefakt të inspektueshëm. Ky është një pozicionim i ndërgjegjshëm *kundër* gamifikimit të zbrazët dhe *në favor* të kompetencës së dëshmuar. Synimi është që ndjesia e përparimit të jetë rrjedhojë e punës reale, jo zëvendësues i saj.

Csikszentmihalyi (1990), me konceptin e *rrjedhës* (flow), tregon se angazhimi optimal ndodh kur sfida përputhet me aftësinë. Adaptimi i vështirësisë në QuestPath synon pikërisht ta mbajë nxënësin afër kësaj gjendjeje.

2.7 Vlerësimi me portofol dhe mësimi përmes të bërit

Tradita konstruktiviste — nga Dewey (1938) te Papert (1980) — argumenton se dija ndërtohet më së miri përmes të bërit dhe prodhimit të artefakteve të prekshme. *Vlerësimi me portofol* (Zubizarreta, 2009) e zbaton këtë: nxënësi grumbullon një koleksion punësh që dëshmojnë rritjen me kalimin e kohës. Roediger dhe Karpicke (2006) shtojnë se vetë akti i rikujtimit dhe i prodhimit forcon mësimin — *efekti i testimit*.

Portofoli i dëshmive në QuestPath është realizimi i drejtpërdrejtë i kësaj tradite: çdo mision i përfunduar kontribuon një artefakt të rishikuar në një portofol të dukshëm, i cili shërben njëkohësisht si dëshmi e rritjes dhe si aset i ndashëm me të tjerët.

2.8 Sistemet adaptive të mësimit dhe pozicionimi i QuestPath

Sistemet adaptive të mësimit synojnë të përshtatin procesin e të mësuarit sipas nevojave dhe progresit të përdoruesit. Qasjet e përdorura ndryshojnë nga rregullat e thjeshta të përcaktuara paraprakisht deri te modelet më komplekse statistikore dhe metodat e mësimit makinerik, siç është gjurmimi i njohurive (*knowledge tracing*). Modelet më komplekse mund të ofrojnë mundësi më të avancuara për përshtatje, por zakonisht kërkojnë sasi më të mëdha të të dhënave dhe mund të jenë më të vështira për t'u interpretuar dhe shpjeguar.

Në **QuestPath**, mekanizmi adaptiv është ndërtuar mbi rregulla të përcaktuara paraprakisht. Përshtatja e misioneve të ardhshme bëhet në bazë të të dhënave të ofruara nga përdoruesi gjatë realizimit të misioneve. Kjo qasje është zgjedhur duke marrë parasysh natyrën e platformës si prototip dhe mungesën e një sasie të madhe të të dhënave që do të nevojitej për përdorimin e metodave të mësimit makinerik. Gjithashtu, përdorimi i rregullave të qarta mundëson që mënyra se si përshtatet rruga e të mësuarit të jetë më e kuptueshme për përdoruesin dhe mentorin.

Tabela në vijim përmbledh parimet kryesore të nxjerra nga literatura dhe i lidh ato me vendimet konkrete të dizajnit të QuestPath.

Parimi (burimi)	Implikimi për dizajnin	Realizimi në QuestPath
Vetërregullimi në SDL (Knowles, 1975)	Nxënësit i nevojiten sinjale të jashtme objektive	Dëshmia e inspektueshme si njësi progresi
Mësimi për zotërim (Bloom, 1968; 1984)	Përparim njësi-pas-njësie me vlerësim të shpeshtë	Misione të vogla ditore me dëshmi për secilin
Reagimi formativ (Hattie & Timperley, 2007)	Reagimi duhet të mbulojë progresin, hendekun, hapin pasues	Fusha të strukturuara: wentWell / missing / improvement / nextAction
Praktika e qëllimshme (Ericsson et al., 1993)	Fokus i përsëritur te dobësitë	Adaptim që forcon pjesën më të dobët

Parimi (burimi)	Implikimi për dizajnin	Realizimi në QuestPath
Vetëefikasiteti (Bandura, 1977)	Suksese të prekshme ndërtojnë besim	XP dhe portofol të bazuar në dëshmi; sinjali i besimit
ZPD (Vygotsky, 1978)	Vështirësia brenda zonës optimale	Vendimi: ngrit nivelin / vazhdo / përsërit më vogël
Ngarkesa kognitive (Sweller, 1988)	Minimizimi i ngarkesës së panevojshme	Misione të vogla, një veprim i qartë për ekran
SDT & kritika e gamifikimit (Ryan & Deci, 2000; Deterding et al., 2011)	Shmangia e shpërblimeve boshe	XP i kushtëzuar nga dëshmia reale
Portofoli & efekti i testimit (Zubizarreta, 2009; Roediger & Karpicke, 2006)	Prodhimi i artefakteve forcon mësimin	Portofol i dukshëm i dëshmive të rishikuara

3 METODOLOGJIA

3.1 Qasja kërkimore: Shkenca e Dizajnit

Ky punim ndjek paradigmën e *Shkencës së Dizajnit* (Design Science Research, DSR). Ndryshe nga kërkimi shpjegues që synon të përshkruajë botën ashtu siç është, DSR prodhon njohuri përmes *ndërtimit dhe vlerësimit të një artefakti* që synon të zgjidhë një problem real. Artefakti qendror këtu është prototipi QuestPath. Njohuria e prodhuar përfshin: modelin konceptual të progresit të bazuar në dëshmi, arkitekturën e zbatueshme, dhe motorin adaptiv të shpjegueshëm.

DSR është zgjedhja e duhur sepse pyetja qendrore e punimit nuk është "a ekziston një fenomen", por "a mund të ndërtohet një artefakt që e mishëron dhe e bën funksional një grup parimesh". Vlerësimi në DSR mund të jetë i shkallëzuar; në këtë fazë ai është i bazuar në prototip dhe skenarë, ndërsa vlerësimi empirik me përdorues është propozuar si punë e ardhshme.

3.2 Procesi i punës

Procesi u zhvillua në gjashtë faza iterative, jo rreptësisht lineare:

1. **Rishikimi i literaturës.** Nxjerrja e parimeve nga pedagogjia dhe psikologjia e mësimnxënies, të cilat u përkthyen në kërkesa.
2. **Analiza e kërkesave.** Përcaktimi i profileve të përdoruesve, rasteve të përdorimit dhe kërkesave funksionale/jofunksionale.
3. **Dizajni i arkitekturës dhe i të dhënave.** Zgjedhja e teknologjive, ndarja e shtresave dhe modelimi i domenit.
4. **Implementimi iterativ.** Zhvillimi i funksionaliteteve të sistemit, autentikimit dhe ndërfaqes së përdoruesit, me përmirësime të vazhdueshme gjatë zhvillimit.
5. **Verifikimi.** Kontrollimi i kodit, testimi i funksionaliteteve kryesore, verifikimi i përshtatjes së ndërfaqes në pajisje të ndryshme dhe kontrollimi i funksionimit të aplikacionit pas publikimit.
6. **Reflektimi dhe dokumentimi.** Paraqitja e kufizimeve të punimit dhe përmbledhja e rezultateve të arritura.

Natyra iterative do të thotë se zbulimet në faza të mëvonshme (p.sh. mbingarkesa horizontale në pajisje mobile e zbuluar gjatë verifikimit) shkaktuan rikthime te implementimi.

3.3 Qasja e prototipimit

Gjatë zhvillimit të *QuestPath* është krijuar një prototip funksional i platformës. Qëllimi nuk ka qenë vetëm të paraqitet ideja e sistemit, por të zhvillohet një aplikacion që funksionon dhe që mund të përmirësohet më tej në të ardhmen. Funksionalitetet e përshkruara në këtë punim janë implementuar dhe mund të testohen në platformë.

Gjatë zhvillimit është përdorur *Git* për ruajtjen dhe menaxhimin e versioneve të kodit. Aplikacioni është ndarë në pjesë të ndryshme për ta bërë kodin më të organizuar dhe më të lehtë për mirëmbajtje. Janë përdorur gjithashtu të dhëna testuese për kontrollimin e funksionaliteteve, ndërsa baza e të dhënave *PostgreSQL* është konfiguruar veçmas nga pjesët e tjera të aplikacionit.

3.4 Vlerësimi dhe testimi i bazuar në skenarë

Në mungesë të një studimi me përdorues realë, funksionimi i platformës është kontrolluar përmes skenarëve të përdorimit. Këta skenarë paraqesin veprimet që një *përdorues* ose *mentor* mund të kryejë gjatë përdorimit të platformës. Për shembull, një përdorues hyn në platformë, shikon misionin e ditës, dorëzon dëshminë dhe reflektimin e tij, merr reagim dhe vazhdon me hapat e ardhshëm. Secili skenar është testuar në prototip për të kontrolluar nëse funksionalitetet kryesore punojnë siç duhet nga fillimi deri në fund.

Testimi përmes këtyre skenarëve ndihmon në verifikimin e funksionimit të platformës, por nuk e zëvendëson vlerësimin me përdorues realë. Përmes këtij testimi kontrollohet nëse funksionalitetet e sistemit punojnë siç është paraparë, por nuk vlerësohet ndikimi i platformës në procesin e të mësuarit.

3.5 Mjetet dhe mjedisi

Zhvillimi i aplikacionit është realizuar duke përdorur *Node.js* dhe *pnpm* për menaxhimin e paketave. Për ruajtjen dhe menaxhimin e të dhënave është përdorur *PostgreSQL*, ndërsa *Docker* është përdorur për konfigurimin e mjedisit të bazës së të dhënave.

Për publikimin dhe menaxhimin e aplikacionit është përdorur platforma *Coolify*, e cila mundëson përditësimin automatik të aplikacionit pas ndryshimeve në kod. Për kontrollimin e cilësisë dhe funksionimit të aplikacionit janë përdorur *ESLint*, mjetet e *Next.js* dhe *Playwright*,

përfshirë testimin e funksionaliteteve dhe paraqitjes së aplikacionit në madhësi të ndryshme të ekranit.

3.6 Konsideratat etike

Në kuadër të këtij punimi nuk është realizuar ndonjë studim me përdorues realë dhe nuk janë mbledhur të dhëna personale. Të dhënat e përdorura gjatë zhvillimit dhe testimit të platformës janë të dhëna testuese dhe nuk u përkasin personave realë.

Gjatë zhvillimit të platformës është marrë parasysh edhe mbrojtja e të dhënave të përdoruesve. Fjalëkalimet nuk ruhen në formën e tyre të zakonshme, por ruhen në mënyrë të sigurt përmes procesit të hashimit. Gjithashtu, të dhënat konfidenciale të konfigurimit të aplikacionit ruhen veçmas dhe nuk përfshihen në kodin burimor.

4 KËRKESAT DHE ANALIZA E SISTEMIT

Ky kapitull paraqet kërkesat kryesore të sistemit dhe analizon funksionalitetet që duhet të ofrojë platforma *QuestPath*. Kërkesat janë përcaktuar duke u bazuar në qëllimin e platformës dhe në funksionalitetet e implementuara në prototip.

4.1 Palët e interesit dhe përdoruesit e sistemit

Sistemi ka dy lloje kryesore të përdoruesve: *përdoruesin*, i cili ndjek misionet dhe dorëzon dëshmi për punën e realizuar, dhe *mentorin*, i cili shqyrton punën e dorëzuar dhe jep komente. Tabela e mëposhtme paraqet përdoruesit e sistemit dhe nevojat kryesore të tyre.

Personi	Përshkrimi	Nevojat kryesore
Përdoruesi	Personi që përdor platformën për të zhvilluar aftësi dhe për të realizuar misionet e caktuara.	Të shohë misionin e radhës, të dorëzojë dëshmi për punën e realizuar dhe të përcjellë progresin e tij.
Mentori	Personi që shqyrton dëshmitë e dorëzuara dhe jep vlerësime dhe komente.	Të ketë mundësi të shqyrtojë punën e dorëzuar, ta vlerësojë dhe të japë komente për përdoruesin.

Personi	Përshkrimi	Nevojat kryesore
Administratori	Personi përgjegjës për menaxhimin dhe mirëmbajtjen e sistemit.	Të menaxhojë sistemin dhe të sigurojë funksionimin e rregullt dhe sigurinë bazë të tij.

Përdoruesi është në qendër të platformës dhe duhet ta ketë të qartë çfarë duhet të bëjë më pas. Mentori duhet ta ketë të lehtë të shohë dhe të vlerësojë punën e dorëzuar.

4.2 Kërkesat funksionale

Tabela e mëposhtme liston kërkesat funksionale (FR), të gjitha të realizuara në prototip.

ID	Kërkesa funksionale	Statusi
FR1	Përdoruesi mund të hyjë në sistem me email dhe fjalëkalim	I implementuar
FR2	Sistemi ruan një sesion pas hyrjes së suksesshme	I implementuar
FR3	Përdoruesi mund të dalë nga sistemi	I implementuar
FR4	Përdoruesi i ri zgjedh një rrugë mësimi dhe e kalibron atë	I implementuar
FR5	Paneli kryesor shfaq misionin e ditës dhe progresin	I implementuar
FR6	Përdoruesi sheh hartën e misioneve me gjendjet e tyre	I implementuar
FR7	Përdoruesi mund të shohë detajet dhe kërkesat e një misioni	I implementuar
FR8	Përdoruesi mund të dorëzojë dëshminë, reflektimin, nivelin e besimit dhe shënimet	I implementuar
FR9	Sistemi kontrollon nëse të dhënat e kërkuara janë plotësuar para dorëzimit	I implementuar
FR10	Sistemi jep reagim pas dorëzimit të misionit	I implementuar
FR11	Sistemi përshtat hapat e ardhshëm në bazë të të dhënave të dorëzuara	I implementuar
FR12	Dorëzimi i parë jep XP, përditëson nivelin dhe streak-un	I implementuar
FR13	Dorëzimi përditëson progresin e aftësive përkatëse	I implementuar
FR14	Sistemi jep arritje sipas misioneve të përfunduara	I implementuar
FR15	Përdoruesi sheh pemën e aftësive	I implementuar

ID	Kërkesa funksionale	Statusi
FR16	Përdoruesi mund të shohë dëshmitë e dorëzuara dhe të shqyrtuara	I implementuar
FR17	Përdoruesi mund të shohë profilin dhe progresin e tij	I implementuar
FR18	Mentori mund të shqyrtojë dhe vlerësojë punën e dorëzuar	Pjesërisht i implementuar
FR19	Sistemi përmban modelin për versionin falas dhe versionin Pro	I paraparë, por jo i implementuar plotësisht

4.3 Kërkesat jofunksionale

ID	Kërkesa jofunksionale	Si adresohet
NFR1	Siguria e fjalëkalimeve	Fjalëkalimet ruhen në mënyrë të sigurt duke përdorur bcrypt.
NFR2	Siguria e sesionit	Sesioni i përdoruesit mbrohet për të parandaluar ndryshimet ose qasjen e paautorizuar.
NFR3	Performanca e sistemit	Aplikacioni është ndërtuar në mënyrë që faqet dhe të dhënat të ngarkohen sa më shpejt.
NFR4	Responsiviteti	Ndërfaqja përshtatet për përdorim në web dhe pajisje mobile.
NFR5	Qasshmëria	Kontrast i synuar AA, navigim me tastierë, gjendje fokusi, ngjyra jo si sinjal i vetëm (W3C, 2018).
NFR6	Mirëmbajtja e sistemit	Ndarje e qartë e shtresave, TypeScript, modele të tipizuara.
NFR7	Menaxhimi i bazës së të dhënave	Ndryshimet në strukturën e bazës së të dhënave menaxhohen dhe ruhen në mënyrë të organizuar.
NFR8	Mbrojtja e të dhënave të konfigurimit	Të dhënat konfidenciale të konfigurimit ruhen veçmas dhe nuk përfshihen në kodin burimor.
NFR9	Publikimi dhe funksionimi i aplikacionit	Për publikimin dhe menaxhimin e aplikacionit përdoren Docker dhe Coolify.

4.4 Rastet kryesore të përdorimit

ID	Rasti i përdorimit	Aktori	Rrjedha
UC1	Hyrje në sistem	Përdoruesi	Vendos emailin dhe fjalëkalimin. Sistemi kontrollon të dhënat dhe hap panelin kryesor.

ID	Rasti i përdorimit	Aktori	Rrjedha
UC2	Zgjedhja e rrugës së të mësuarit	Përdoruesi	Zgjedh rrugën e të mësuarit dhe përcakton kohën, nivelin, vështirësinë dhe afatin. Sistemi krijon rrugën përkatëse.
UC3	Dorëzimi i dëshmisë	Përdoruesi	Hap misionin, vendos dëshminë dhe reflektimin. Sistemi kontrollon dhe ruan të dhënat dhe më pas jep reagim
UC4	Shikimi i reagimit dhe hapit të ardhshëm	Përdoruesi	Shikon reagimin e marrë dhe misionin e ardhshëm të përshtatur nga sistemi.
UC5	Rishikim i dëshmisë	Mentori	Shqyrton dhe vlerëson punën e dorëzuar, jep koment dhe e miraton ose kërkon ndryshime.
UC6	Shikimi i dëshmive	Përdoruesi	Shikon dëshmitë e dorëzuara dhe të shqyrtuara më parë.

4.4.1 Skenar i detajuar

Në këtë skenar, përdoruesi hap misionin dhe dorëzon dëshminë për punën e realizuar. Ai vendos lidhjen e dëshmisë, shkruan një reflektim dhe tregon nivelin e besimit nga 1 deri në 5. Para dorëzimit, sistemi kontrollon nëse të dhënat e kërkuara janë plotësuar. Pas dorëzimit, të dhënat ruhen, përdoruesi merr reagim dhe përditësohet progresi i tij në platformë.

4.5 Analiza e domenit dhe fjalori

- **Rruga e të mësuarit:** paraqet rrugën që ndjek përdoruesi për të arritur një qëllim të caktuar.
- **Modeli i rrugës:** përmban strukturën dhe misionet që përdoren për krijimin e një rruge të të mësuarit.
- **Rruga e përdoruesit:** rruga e zgjedhur nga përdoruesi, e përshtatur sipas nivelit dhe qëllimeve të tij.
- **Misioni:** një detyrë që përdoruesi duhet ta realizojë dhe për të cilën duhet të dorëzojë dëshmi.
- **Dëshmi / Dorëzim:** artefakti dhe reflektimi që nxënësi dorëzon.
- **Reagim:** vlerësimi ose komenti që përdoruesi merr pas dorëzimit të punës.
- **Aftësi:** një aftësi që përdoruesi zhvillon gjatë realizimit të misioneve.
- **Arritje:** një shenjë e fituar për momente kyçe.

4.6 Kufizimet dhe supozimet

Prototipi aktual ka disa kufizime. Përdoruesi mund të ketë një rrugë aktive të të mësuarit, ndërsa pjesa e mentorit përdor të dhëna testuese. Këto funksionalitete mund të përmirësohen në të ardhmen.

5 ARKITEKTURA

5.1 Pamje e përgjithshme dhe stili arkitektonik

QuestPath është ndërtuar duke përdorur *Next.js* dhe modelin App Router (Vercel, 2026). Aplikacioni është organizuar në disa pjesë, ku pjesa kryesore e përpunimit të të dhënave kryhet në server. Kjo përfshin autentikimin e përdoruesve, ruajtjen dhe marrjen e të dhënave, si dhe funksionet kryesore të sistemit. Ndërfaqja e përdoruesit shfaq të dhënat dhe mundëson ndërveprimin me platformën.

QuestPath është zhvilluar si një aplikacion i vetëm dhe jo si një sistem i ndarë në mikroshërbime. Kjo qasje është më e përshtatshme për këtë prototip, pasi e bën zhvillimin, menaxhimin dhe mirëmbajtjen e aplikacionit më të thjeshtë. Në të njëjtën kohë, kodi është organizuar në pjesë të ndara sipas funksioneve të sistemit. Gjithashtu, komunikimi me serverin realizohet përmes Server Actions dhe jo përmes një API-je të veçantë REST, stil i cili u formalizua nga Fielding (2000).

5.2 Teknologjitë e përdorura dhe arsyetimi i përzgjedhjes

Shtresa	Teknologjia	Arsyetimi
Web Framework	Next.js 16 (App Router) (Vercel, 2026)	Renderim në server, Server Actions, rrugëzim i bazuar në skedarë
Gjuha	TypeScript (TypeScript, 2026)	Tipizim statik, zbulim më i lehtë i gabimeve dhe mirëmbajtje më e lehtë e kodit
UI	React 19 (Meta Open Source, 2026) + Tailwind CSS 4 (Tailwind Labs, 2026)	Komponentë të ripërdorshëm; stilizim i shpejtë dhe konsistent
Ikona	lucide-react	Grup ikonash i lehtë dhe konsistent
ORM	Prisma 7 (+ adapteri pg) (Prisma, 2026)	Skemë e tipizuar, migrime, klient i gjeneruar
Baza e të dhënave	PostgreSQL (PostgreSQL Global Development Group, 2026)	Bazë relacionale për ruajtjen dhe menaxhimin e të dhënave
Autentikimi	bcryptjs + cookie	Mbrojtja e fjalëkalimeve dhe menaxhimi i sesionit të përdoruesit

Shtresa	Teknologjia	Arsyetimi
Mjedisi i aplikacionit	Docker (Docker Inc., 2026)	Krijimi i një mjedisi të njëjtë për ekzekutimin e aplikacionit
Publikimi	Coolify (Coolify, 2026)	Publikimi dhe përditësimi automatik i aplikacionit

Zgjedhja e këtyre teknologjive prioritizon qëndrueshmërinë, tipizimin dhe kompleksitetin e ulët operacional. Asnjë prej tyre nuk është eksperimentale; të gjitha janë standarde industrie me dokumentim të pasur.

5.3 Diagrami i arkitekturës së sistemit

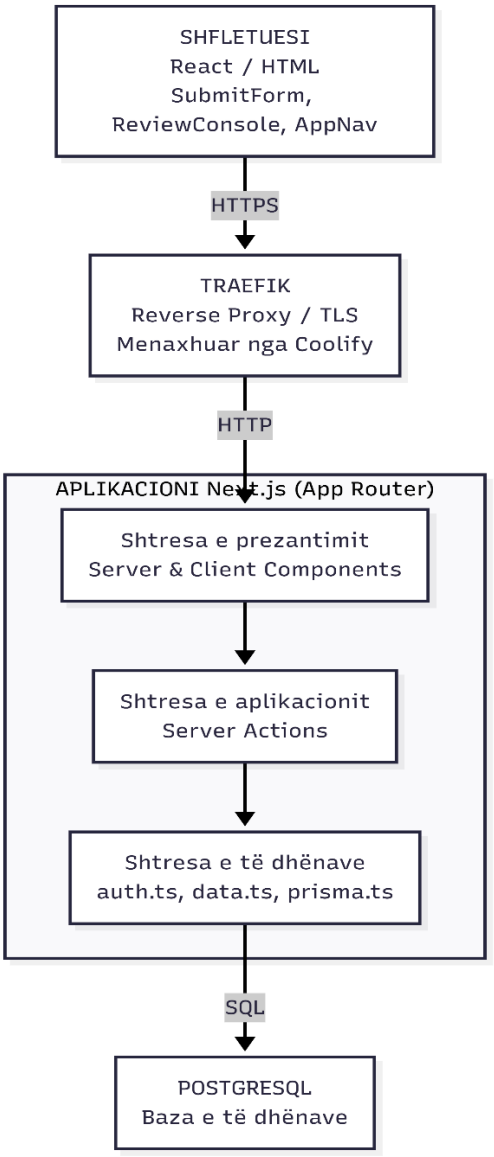

Figura 6. *Diagrami i arkitekturës së sistemit.*

5.4 Arkitektura App Router dhe komponentët Server/Client

Një pjesë e rëndësishme e arkitekturës së aplikacionit është ndarja *ndërmjet Server Components dhe Client Components (Meta Open Source, 2026). Server Components* përdoren për pjesët që përpunohen në server dhe për marrjen e të dhënave, ndërsa *Client Components* përdoren në pjesët ku nevojitet ndërveprim me përdoruesin.

Në aplikacion përdoren këto Client Components:

- **SubmitForm** – përdoret për plotësimin dhe dorëzimin e dëshmisë.
- **ReviewConsole** – përdoret nga mentori për shqyrtimin dhe vlerësimin e punës së dorëzuar.
- **AppNav** – përdoret për navigimin ndërmjet faqeve të aplikacionit.

Pjesët e tjera të aplikacionit përdorin kryesisht Server Components. Kjo ndarje ndihmon në organizimin e aplikacionit dhe zvogëlon sasinë e JavaScript-it që duhet të ngarkohet në shfletues.

5.5 Shtresa e qasjes në të dhëna

Qasja në të dhëna është e centralizuar në tre module:

- `prisma.ts` — menaxhon lidhjen e aplikacionit me bazën e të dhënave përmes Prisma (Prisma, 2026).
- `data.ts` — përmban funksionet që përdoren për marrjen e të dhënave të përdoruesit dhe misioneve.
- `actions.ts` — përmban veprimet që ndryshojnë të dhënat, si hyrja dhe dalja nga sistemi, fillimi i një rruge të të mësuarit dhe dorëzimi i dëshmisë.

Kjo ndarje e bën kodin më të organizuar dhe ndan funksionet për marrjen e të dhënave nga ato që bëjnë ndryshime në sistem. Kjo ndarje ndjek parimin e shtresëzimit dhe të ndarjes së përgjegjësive në arkitekturën e aplikacioneve (Fowler, 2002).

5.6 Harta e navigimit dhe arkitektura e informacionit

Këtu paraqitet harta e navigimit. Aplikacioni ndahet në një zonë publike dhe një zonë të mbrojtur ku çdo rrugë kërkon një përdorues të vlefshëm.

PUBLIKE

/login ——————————▶ *Hyrja në sistem*

E MBROJTUR

/onboarding ——————————▶ *Zgjedhja dhe konfigurimi i rrugës së të mësuarit*

/dashboard ——————————▶ *Misioni i ditës dhe progresi*

/map ——————————▶ *Harta e misioneve*

/quests/[id] ——————————▶ *Detajet e misionit*

/submit/[id] ——————————▶ *Dorëzimi i dëshmisë*

/feedback/[id] ——————————▶ *Reagimi pas dorëzimit*

/skills ——————————▶ *Aftësitë e përdoruesit*

/portfolio ——————————▶ *Dëshmitë e dorëzuara*

/profile ——————————▶ *Profili dhe progresi*

/review ——————————▶ *Shqyrtimi i dëshmive nga mentori*

Figura 10. *Harta e navigimit dhe arkitektura e informacionit.*

Navigimi në aplikacion realizohet përmes AppNav. Në kompjuter, menyja paraqitet në anën e majtë, ndërsa në pajisje mobile paraqitet në pjesën e poshtme të ekranit. Faqja aktive dallohet me ngjyrë blu, në përputhje me parimin e dukshmërisë së gjendjes së sistemit (Nielsen, 1994).

5.7 Arkitektura e publikimit të aplikacionit

Aplikacioni *QuestPath* është publikuar duke përdorur *Docker* dhe *Coolify*. Baza e të dhënave *PostgreSQL* është konfiguruar veçmas dhe lidhet me aplikacionin për ruajtjen dhe marrjen e të dhënave. Kjo mënyrë e organizimit ndihmon që aplikacioni dhe baza e të dhënave të funksionojnë në mënyrë të pavarur nga shërbimet e tjera në server.

6 IMPLEMENTIMI

Ky kapitull përshkruan mënyrën se si është implementuar platforma QuestPath. Paraqiten pjesët kryesore të aplikacionit, organizimi i kodit dhe funksionalitetet e zhvilluara.

6.1 Struktura e projektit

Kodi i aplikacionit është organizuar duke përdorur strukturën e Next.js App Router:

```
src/
├── app/
│   ├── layout.tsx              # struktura kryesore e aplikacionit
│   ├── globals.css             # stilet e aplikacionit
│   ├── page.tsx                # faqja fillestare
│   ├── login/page.tsx          # faqja e hyrjes
│   └── (app)/                  # faqet pas kyçjes
│       ├── layout.tsx          # struktura dhe navigimi
│       ├── dashboard/page.tsx
│       ├── onboarding/page.tsx
│       ├── map/page.tsx
│       ├── quests/[id]/page.tsx
│       ├── submit/[id]/page.tsx + SubmitForm.tsx
│       ├── feedback/[id]/page.tsx
│       ├── skills/page.tsx
│       ├── portfolio/page.tsx
│       ├── profile/page.tsx
│       └── review/page.tsx   #shqyrtimi nga mentori
├── components/                 # komponentët e ndërfaqes
└── lib/                        # funksionet dhe lidhja me të dhënat
```

Tabela e mëposhtme përmbledh rrugët dhe komponentët përgjegjës.

Rruga	Lloji kryesor	Komponenti përgjegjës
`/login`	Server Component	`login/page.tsx` + `loginAction`
`/onboarding`	Server Component	`onboarding/page.tsx` + `startPathAction`
`/dashboard`	Server Component	`dashboard/page.tsx` (përdor `getAppState`)
`/map`	Server Component	`map/page.tsx`

Rruga	Lloji kryesor	Komponenti përgjegjës
`/quests/[id]`	Server Component	`quests/[id]/page.tsx` (përdor `getQuestForUser`)
`/submit/[id]`	Server + Client	`page.tsx` + `SubmitForm.tsx` (`useActionState`)
`/feedback/[id]`	Server Component	`feedback/[id]/page.tsx`
`/skills`	Server Component	`skills/page.tsx`
`/portfolio`	Server Component	`portfolio/page.tsx`
`/profile`	Server Component	`profile/page.tsx`
`/review`	Server + Client	`page.tsx` + `ReviewConsole.tsx`

6.2 Autentikimi dhe sesionet

Autentikimi i përdoruesve është realizuar në skedarin `src/lib/auth.ts`. Fjalëkalimet kontrollohen duke përdorur bcrypt, ndërsa pas hyrjes së suksesshme krijohet një sesion që ruhet përmes një cookie. Kjo i mundëson sistemit të kontrollojë nëse përdoruesi është i kyçur.

```
function sign(userId: string) {
  return crypto.createHmac("sha256", authSecret()).update(userId).digest("hex");
}

export async function setSession(userId: string) {
  const jar = await cookies();
  jar.set(COOKIE_NAME, `${userId}.${sign(userId)}`, {
    httpOnly: true,
    sameSite: "lax",
    secure: process.env.NODE_ENV === "production",
    path: "/",
    maxAge: 60 * 60 * 24 * 30,
  });
}
```

Kur lexohet sesioni, nënshkrimi rikrijohet dhe krahasohet me atë të ruajturin përmes `crypto.timingSafeEqual`, e cila parandalon sulmet me kohëmatje. Krahasimi kryhet vetëm pasi konfirmohet barazia e gjatësive:

```
export async function currentUser() {
  const jar = await cookies();
  const token = jar.get(COOKIE_NAME)?.value;
```

```
  if (!token) return null;

  const [userId, signature] = token.split(".");
  const expected = sign(userId);
  if (!userId || !signature || signature.length !== expected.length ||
      !crypto.timingSafeEqual(Buffer.from(signature), Buffer.from(expected))) {
    return null;
  }
  return prisma.user.findUnique({ where: { id: userId } });
}
```

Funksioni *requireUser* kontrollon nëse përdoruesi është i kyçur në sistem. Nëse nuk është i kyçur, ai r idrejtohet në faqen /login. Ky kontroll bëhet në pjesët e aplikacionit që kërkojnë autentikim, du ke parandaluar qasjen e paautorizuar.

6.3 Server Actions dhe përpunimi i të dhënave

Veprimet kryesore të sistemit janë të organizuara në skedarin `src/lib/actions.ts`. Ato përdoren për hyrjen dhe daljen nga sistemi, krijimin e rrugës së të mësuarit dhe dorëzimin e dëshmive. Tabela e mëposhtme paraqet veprimet kryesore dhe funksionin e tyre.

Veprimi	Hyrjet	Efektet kryesore
loginAction	email, fjalëkalim	Validon kredencialet, vendos sesionin, ridrejton te dashboard
logoutAction	—	Pastron sesionin, ridrejton te login
startPathAction	shabllon, kalibrim	Çaktivizon rrugët e mëparshme, krijon UserPath të ri
submitProofAction	mision, link, reflektim, besim, shënime	Validon, ruan dorëzimin, gjeneron reagim, jep XP/aftësi/arritje
ensureOnboarding	—	Kontrollon nëse përdoruesi ka një rrugë aktive dhe, nëse jo, e dërgon te konfigurimi fillestar.

Hyrja normalizon email-in dhe e trajton dështimin njësoj pavarësisht nëse përdoruesi ekziston apo jo, duke shmangur zbulimin e ekzistencës së llogarive:

```
export async function loginAction(formData: FormData) {
  const email = String(formData.get("email") ?? "").toLowerCase().trim();
  const password = String(formData.get("password") ?? "");
  const user = await prisma.user.findUnique({ where: { email } });

  if (!user || !(await verifyPassword(password, user.passwordHash))) {
    redirect("/login?error=invalid");
  }
  await setSession(user.id);
  redirect("/dashboard");
}
```

Funksioni *submitProofAction* përdoret për dorëzimin e dëshmisë së një misioni. Sistemi kontr ollon nëse është vendosur lidhja e dëshmisë, nëse reflektimi ka të paktën 20 karaktere dhe nës e niveli i besimit është nga 1 deri në 5. Pas kontrollit, dorëzimi ruhet dhe përditësohet progresi i përdoruesit. Pikët XP dhe arritjet jepen vetëm gjatë dorëzimit të parë të misionit.

```
if (!proofUrl) return { error: "Add a proof link so your mentor can inspect the work." };
if (reflection.length < 20) return { error: "Add a short reflection — a sentence or two…" };
// … upsert i dorëzimit dhe i reagimit …
if (!existing) {
  // jep XP për aftësitë, përditëson nivelin/streak, numëron misionet e përfunduara,
  // përditëson UserPath.currentDay dhe jep arritjen sipas quest.kind
}
```

Nëse i njëjti mision dorëzohet përsëri, sistemi përditëson dëshminë dhe reagimin, por nuk i jep përsëri pikët XP dhe arritjet.

6.4 Marrja e të dhënave

Marrja e të dhënave realizohet përmes skedarit src/lib/data.ts. Funksioni getAppState merr të d hënat kryesore të përdoruesit, si progresin, arritjet, rrugën aktive, misionet dhe dëshmitë e dor ëzuara. Sistemi përcakton si “mision të ditës” misionin e parë që përdoruesi ende nuk e ka për funduar.

```
const done = new Set(submissions.map((s) => s.questId));
const todayQuest =
  activePath.template.quests.find((q) => !done.has(q.id)) ??
  activePath.template.quests[activePath.template.quests.length - 1] ??
  null;
```

Funksioni `getQuestForUser` shton një kontroll qasjeje: një mision është i arritshëm vetëm nëse nxënësi ka një rrugë mbi të njëjtin shabllon. Ky kontroll parandalon qasjen e padrejtë në misione të rrugëve që nxënësi nuk i ka nisur.

6.5 Ndërfaqja e përdoruesit

Ndërfaqja e *QuestPath* është dizajnuar me një pamje të thjeshtë dhe të qartë. Në aplikacion përdoren kryesisht ngjyra blu, e bardhë dhe gri, ndërsa elementet e rëndësishme dallohen me ngjyra të veçanta. Dizajni ndjek parimet e njohura të përdorshmërisë: dukshmëri e gjendjes së sistemit, konsistencë, gjuhë e kuptueshme për përdoruesin dhe parandalim i gabimeve (Nielsen, 1994).

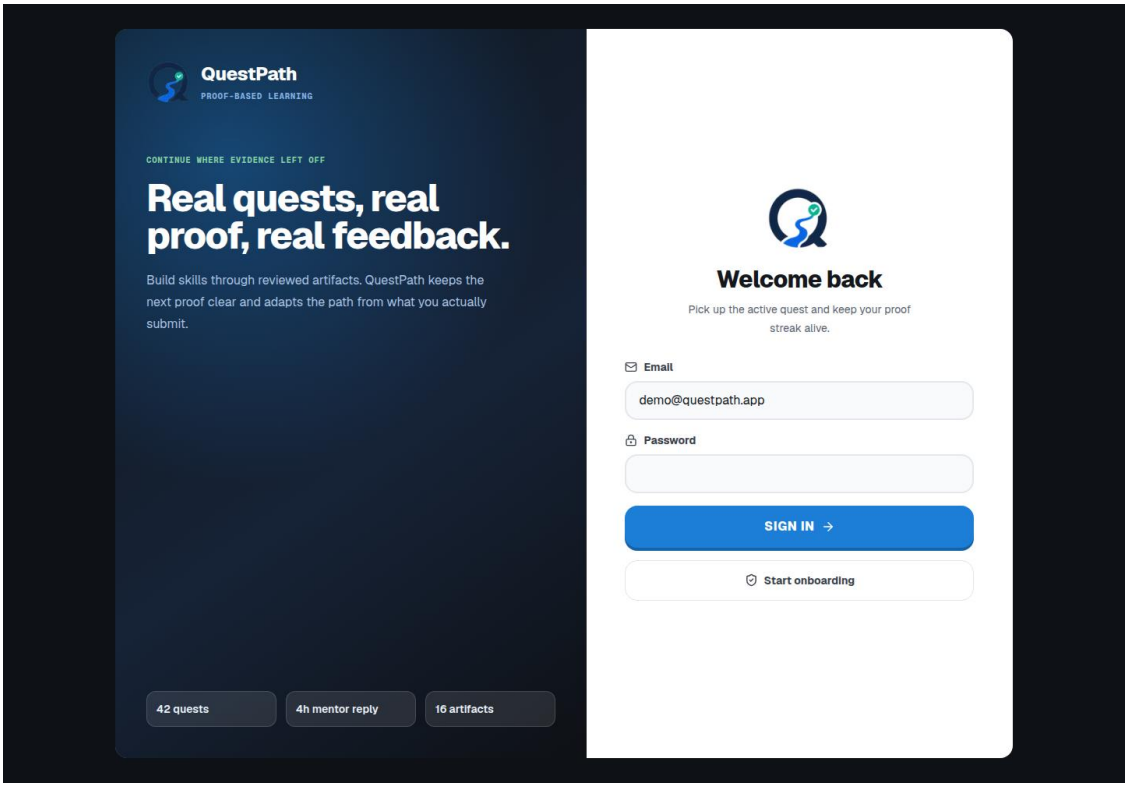

Figura 1. *Pamja e faqes së hyrjes (Login).*

Përdoruesi mund të shohë misionin aktual, reagimin e marrë dhe misionet e ardhshme. Gjithashtu, shfaqen pikët XP, dëshmitë e dorëzuara dhe progresi i përdoruesit.

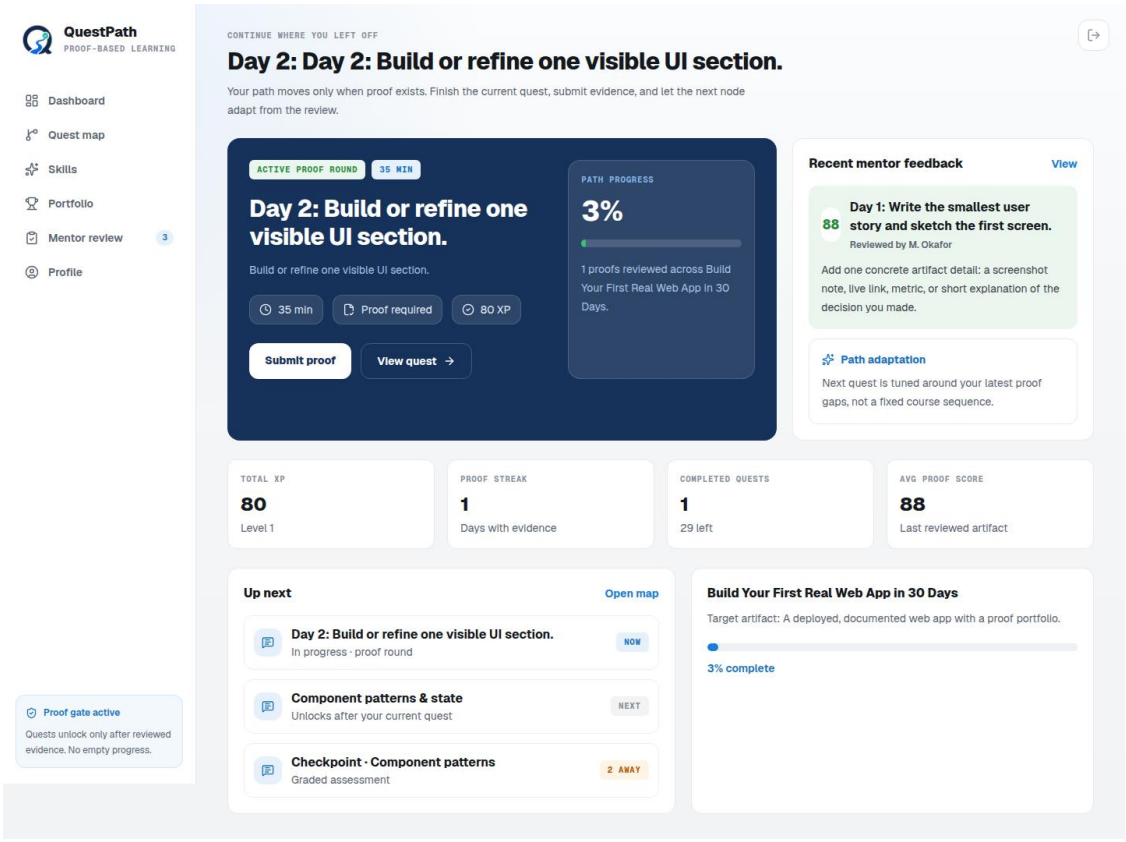

Figura 2. *Paneli kryesor (Dashboard)-pamja në desktop.*

Në pajisje mobile, përmbajtja përshtatet me madhësinë e ekranit dhe menyja e navigimit paraqitet në pjesën e poshtme. Paraqitja është testuar për të siguruar funksionimin e duhur në ekrane të ndryshme.

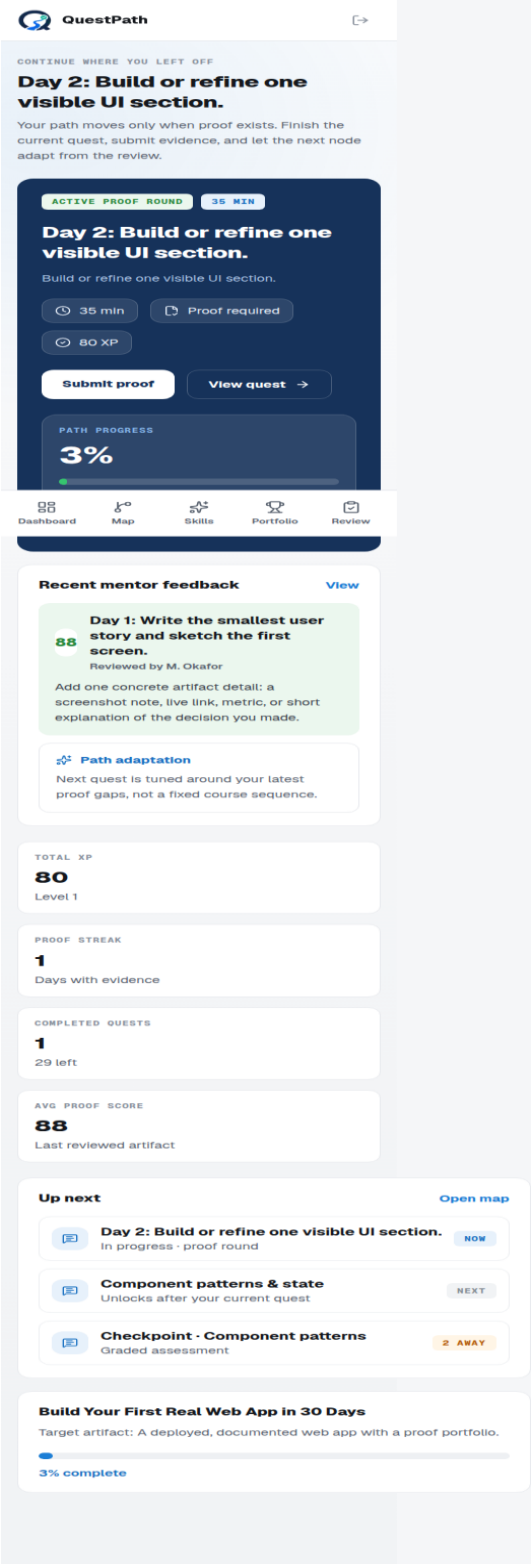

Figura 3. *Paneli kryesor (Dashboard) - pamja në pajisje mobile*

Paraqitja e aplikacionit është rregulluar që të përshtatet si duhet në ekrane të ndryshme.

6.6 Formulari i dorëzimit dhe gjendja interaktive

SubmitForm përdoret për dorëzimin e dëshmisë së një misioni. Formulari kontrollon nëse përdoruesi i ka plotësuar të dhënat e kërkuara para dorëzimit. Përdoruesi zgjedh nivelin e besimit nga 1 deri në 5 dhe plotëson fushat e nevojshme. Nëse mungon ndonjë e dhënë, sistemi shfaq një mesazh dhe nuk lejon dorëzimin derisa formulari të plotësohet siç duhet. Kjo sjellje i përgjigjet parimit të parandalimit të gabimeve (Nielsen, 1994).

6.7 Cikli i përpunimit të një kërkese

Gjatë dorëzimit të një dëshmie, sistemi kontrollon dhe ruan të dhënat, përditëson progresin dh e më pas shfaq reagimin për përdoruesin.

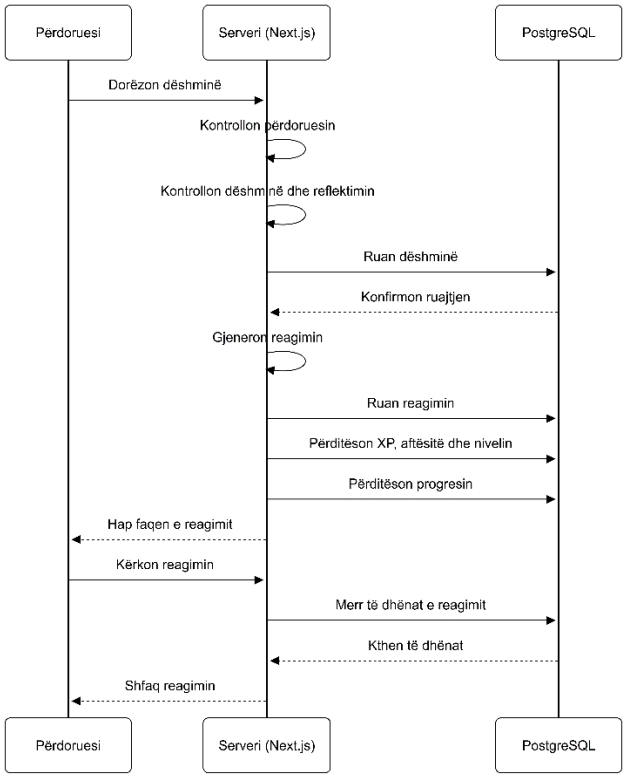

Figura 7. *Cikli i përpunimit të një kërkese përmes Server Actions.*

6.8 Ndërtimi dhe publikimi i aplikacionit

Aplikacioni paketohet me një `Dockerfile` me tre faza për të prodhuar një imazh të vogël dhe të riprodhueshëm (Docker Inc., 2026):

1. **deps** — instalohen paketat e nevojshme të projektit duke përdorur `pnpm`
2. **builder** — përgatitet Prisma dhe ndërtohet aplikacioni Next.js për publikim.

3. **runner** — përgatiten skedarët e nevojshëm dhe aplikacioni niset në server.

Në vijim paraqitet procesi i ndërtimit dhe publikimit të aplikaciAonit.

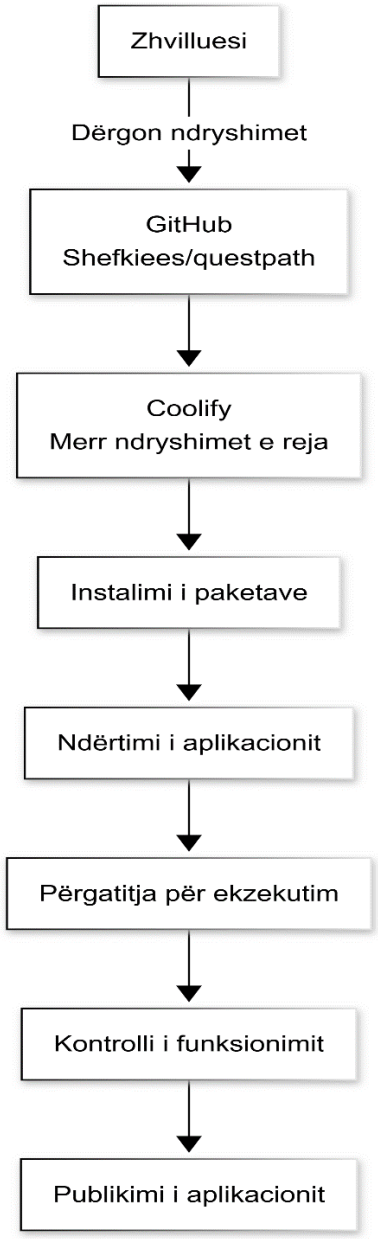

Figura 11. *Procesi i ndërtimit dhe publikimit të aplikacionit.*

Pasi ndryshimet e kodit dërgohen në GitHub, Coolify e përditëson dhe e publikon automatikisht versionin e ri të aplikacionit (Coolify, 2026).

7 MODELI I TË DHËNAVE

7.1 Pamje e përgjithshme

Baza e të dhënave është ndërtuar në PostgreSQL (PostgreSQL Global Development Group, 2026), ndërsa struktura e saj është përcaktuar përmes Prisma (Prisma, 2026). Ajo përbëhet nga njëmbëdhjetë modele dhe tre enume, të organizuara sipas kërkesave funksionale të sistemit.

7.2 Entitetet dhe roli i tyre

Modeli	Roli	Marrëdhëniet kryesore
`User`	Përfaqëson nxënësin dhe ruan të dhënat për XP, nivelin, streak-un dhe planin e tij	1—N me UserPath, QuestSubmission, UserSkillProgress dhe UserAchievement
`PathTemplate`	Përfaqëson strukturën e ripërdorshme të një rruge mësimore	1—N me Quest dhe UserPath
`UserPath`	Përfaqëson një rrugë aktive të caktuar për një nxënës	N—1 me User dhe PathTemplate
`Quest`	Përfaqëson një mision mësimor që kërkon dorëzimin e një dëshmie	N—1 me PathTemplate; 1—N me QuestSubmission
`QuestSubmission`	Përfaqëson dëshminë e dorëzuar nga nxënësi për një mision	N—1 me User dhe Quest; 1—1 me Feedback
`Feedback`	Përmban vlerësimin dhe reagimin e dhënë për dëshminë e dorëzuar	1—1 me QuestSubmission
`Skill`	Përfaqëson një aftësi ose kompetencë që zhvillohet përmes aktiviteteve mësimore	1—N me UserSkillProgress
`UserSkillProgress`	Regjistron progresin e një nxënësi në një aftësi të caktuar	N—1 me User dhe Skill
`Achievement`	Përfaqëson një arritje që mund të fitohet nga nxënësi	1—N me UserAchievement

Modeli	Roli	Marrëdhëniet kryesore
`UserAchievement`	Regjistron arritjet e fituara nga nxënësi dhe kohën e fitimit të tyre	N—1 me User dhe Achievement
`SubscriptionIntent`	Përfaqëson interesimin ose zgjedhjen e përdoruesit për një plan abonimi	N—1 me User, nëse kjo lidhje ekziston në model

7.3 Enumet e domenit

Enum	Vlerat	Përdorimi
`QuestState`	LOCKED, AVAILABLE, COMPLETED, REPEATED	Përcakton gjendjen aktuale të një misioni brenda rrugës mësimore
`QuestKind`	STANDARD, MINI_BOSS, FINAL_BOSS	Përcakton llojin dhe nivelin e rëndësisë së misionit
`SubscriptionPlan`	FREE, PRO_PREVIEW	Përcakton planin e abonimit të përdoruesit

7.4 Diagrami Entitet–Marrëdhënie

Diagrami i paraqitur tregon marrëdhëniet kryesore ndërmjet entiteteve të sistemit. Shenjat 1 d he * tregojnë kardinalitetin e këtyre marrëdhënieve

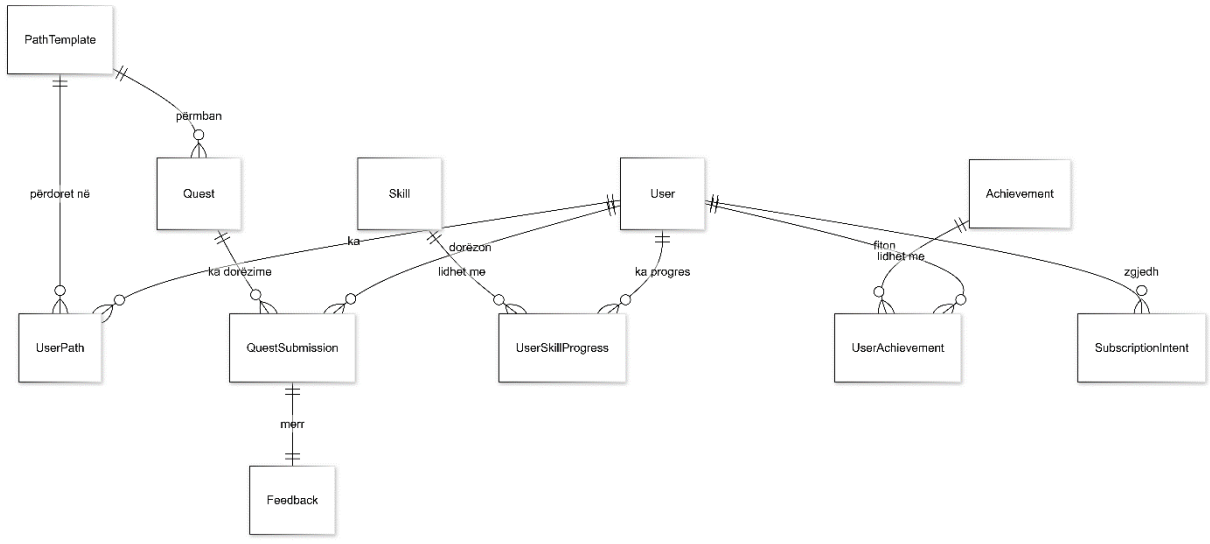

Figura 8. *Diagrami Entitet–Marrëdhënie (ER) i modelit të të dhënave.*

7.5 Marrëdhëniet, integriteti dhe kufizimet

Modeli i të dhënave zbaton integritetin referencial dhe disa kufizime unike për të ruajtur konsistencën e të dhënave:

- QuestSubmission përdor kufizimin unik @@unique([userId, questId]), duke siguruar që një përdorues të ketë vetëm një dorëzim për një mision të caktuar.
- Feedback ka fushën submissionId unike, duke realizuar një marrëdhënie një-me-një me dorëzimin përkatës.
- Quest përdor kufizimin @@unique([templateId, day]), duke siguruar që brenda një shablloni të ketë vetëm një mision për një ditë të caktuar.
- UserSkillProgress dhe UserAchievement përdorin kufizime unike për çiftet (userId, skillId) dhe (userId, achievementId), me qëllim shmangien e të dhënave të dyfishta.
- Fshirja automatike e të dhënave të varura (onDelete: Cascade) siguron që, gjatë fshirjes së një përdoruesi ose të një shablloni, të fshihen edhe të dhënat përkatëse të varura, duke ruajtur konsistencën e bazës së të dhënave.

7.6 Indekset dhe performanca

Struktura e bazës së të dhënave përfshin indekse që ndihmojnë në kërkimin dhe përpunimin më të shpejtë të të dhënave. Modeli UserPath indeksohet sipas fushave *[userId, active]*, për të mundësuar identifikimin e shpejtë të rrugës aktive të një përdoruesi. Modeli Quest indeksohet sipas *[templateId, day]*, ndërsa QuestSubmission sipas *[userId, createdAt]*, për të lehtësuar renditjen kronologjike të dorëzimeve. Fusha *skillKeys* në modelin Quest ruhet si listë vlerash *String[]*, duke mundësuar lidhjen e një misioni me disa aftësi pa pasur nevojë për një tabelë shtesë ndërmjetëse.

7.7 Migrimet dhe të dhënat fillestare

Baza e të dhënave menaxhohet përmes Prisma-s, ndërsa të dhënat fillestare shtohen nga file *prisma/seed.ts*. Në këtë fazë krijohen të dhënat bazë të sistemit, si aftësitë, arritjet, tre modelet e rrugëve mësimore me misionet përkatëse, një përdorues për demonstrim dhe një rrugë aktive. Mënyra e krijimit të misioneve trajtohet më hollësisht më vonë.

8 LOGJIKA ADAPTIVE E BAZUAR NË DËSHMI

Ky kapitull paraqet pjesën kryesore të qasjes së përdorur në QuestPath. Në të shpjegohet mënyra se si dështmitë e dorëzuara nga përdoruesi shndërrohen në progres dhe si sistemi përshtet hapat e mëtejshëm bazuar në rezultatet e tij. Kjo logjikë mbështetet në rregulla të përcaktuara paraprakisht dhe jo në modele të mësimit makinerik, me qëllim që vendimet e sistemit të jenë sa më të qarta dhe të kuptueshme.

8.1 Modeli i progresit të bazuar në dështmi

Në QuestPath, progresi i nxënësit nuk matet vetëm përmes kohës së kaluar në platformë apo rezultateve të kuizeve, por *mbi bazën e dështmive* që ai dorëzon gjatë realizimit të misioneve. Një mision konsiderohet i përfunduar vetëm pasi nxënësi dorëzon një dështmi konkrete, në formën e një linku, të shoqëruar me një reflektim mbi punën e realizuar.

Dështmia e dorëzuar ka disa role kryesore në sistem:

1. **Përcakton progresin** – XP-ja, niveli dhe zhvillimi i aftësive përditësohen pas përfundimit dhe dorëzimit të misionit.
2. **Shërben si bazë për feedback** – vlerësimi lidhet drejtpërdrejt me dështminë dhe reflektimin e përdoruesit.
3. **Ndikon në hapat e mëtejshëm** – rruga e përdoruesit mund të përshtatet në varësi të cilësisë së punës së dorëzuar dhe vlerësimit të saj.
4. **Ndërton portofolin e nxënësit** – dështmitë e grumbulluara krijojnë një pasqyrë të vazhdueshme të progresit dhe zhvillimit të tij.

Ky model e vendos punën konkrete të përdoruesit në qendër të procesit të të nxënit dhe e lidh progresin me dështmi të dukshme të asaj që ai ka realizuar.

8.2 Gjenerimi i misioneve

Misionet krijohen automatikisht sipas disa rregullave të përcaktuara paraprakisht, të cilat varen nga dita dhe kohëzgjatja e rrugës mësimore. Kjo mënyrë organizimi siguron një përparim gradual, nga detyrat bazë deri te misioni përfundimtar. Tabela e mëposhtme paraqet rregullat kryesore të përdorura për krijimin e misioneve.

Aspekti	Rregulla
`Faza e misionit`	Deri në 25% të rrugës: *Foundation*; deri në 65%: *Build/Practice*; deri para përfundimit: *Proof/Polish*; dita e fundit: *Final Boss*.
Lloji i misionit	Misioni i ditës së fundit është `FINAL_BOSS`. Misione `MINI_BOSS` caktohen rreth mesit të rrugës dhe në afërsisht 80% të saj. Të gjitha misionet e tjera janë `STANDARD`.
Koha e parashikuar	Çdo mision i ditës së shtatë parashikohet të zgjasë rreth 60 minuta, ndërsa misionet e tjera rreth 35 minuta.
Shpërblimi me XP	Misionet `STANDARD` japin 80 XP, ndërsa `MINI_BOSS` dhe `FINAL_BOSS` japin 180 XP.
Aftësitë e përfshira	Për rrugën që lidhet me përgatitjen dhe prezantimin e punimit të diplomës përfshihen aftësi si planifikimi, komunikimi dhe përmirësimi i paraqitjes. Në ditët e tjera, aftësitë ndryshojnë sipas llojit të misionit dhe qëllimit të tij.

Kjo logjikë përdoret për krijimin e tri rrugëve mësimore: “Build Your First Real Web App in 30 Days”, “Become Job-Ready as a Junior Developer in 30 Days” dhe “Prepare and Present Your Presentation Like a Pro in 14 Days”. Gjatë secilës rrugë përfshihen misione të ndërmjetme dhe një mision përfundimtar, të cilat shërbejnë si pika kontrolli për të vlerësuar progresin e nxënësit. Kjo qasje mbështetet në parimin e përvetësimit gradual të njohurive, ku aftësitë zhvillohen hap pas hapi dhe zbatohen në detyra më të plota (Bloom, 1968).

8.3 Treguesit kryesorë për përshtatjen e sistemit

Vendimi i sistemit për përshtatjen e hapit të ardhshëm bazohet në dy tregues që merren në *momentin e dorëzimit*, pa pasur nevojë për të dhëna të mëparshme ose modele të trajnuara:

1. **Niveli i besimit (1–5).** Nxënësi vlerëson vetë sa i sigurt ndihet për punën që ka dorëzuar. Ky vlerësim jep një tregues të drejtpërdrejtë për perceptimin që nxënësi ka ndaj aftësive të veta dhe ndihmon sistemin të përshtatë nivelin e vështirësisë (Bandura, 1977).
2. **Thellësia e reflektimit.** Reflektimi i shkruar nga nxënësi përdoret si një tregues i angazhimit të tij me detyrën dhe me procesin e të nxënit. Një reflektim më i zhvilluar mund të tregojë përfshirje më të madhe në proces, megjithëse ky tregues nuk është gjithmonë i saktë.

Kombinimi i këtyre *dy treguesve* i mundëson sistemit të marrë një vendim më të përshtatshëm për hapin e ardhshëm, sesa duke u mbështetur vetëm në njërin prej tyre.

.

8.4 Mekanizmi i reagimit dhe përshtatjes adaptive

Funksioni *mockFeedback* në *src/lib/actions.ts* gjeneron një reagim të ndarë në pesë pjesë, të bazuara në modelin e Hattie dhe Timperley (2007).

Fusha e reagimit	Pyetja udhëzuese	Përmbajtja
`wentWell`	Si po shkoj?	Evidenton aspektet që janë realizuar mirë në dëshminë e dorëzuar dhe merr parasysh nivelin e reflektimit.
`missing`	Çfarë mungon?	Tregon se çfarë mund të plotësohet ose të bëhet më e qartë në dëshminë e dorëzuar.
`improvement`	Si mund të përmirësohet?	Jep një sugjerim konkret për përmirësimin e punës.
`nextAction`	Cili është hapi i ardhshëm?	Sugjeron veprimin e radhës duke marrë parasysh nivelin e besimit të përdoruesit.
`adaptationDecision`	Si vazhdon rruga mësimore?	Përcakton mënyrën se si sistemi përshtet hapin e ardhshëm të përdoruesit.

Vendimi adaptiv prodhohet nga rregullat e mëposhtme:

Kushti	Vendimi i sistemit
Besimi ≥ 4 dhe reflektimi ka më shumë se 160 karaktere	Ngritje e nivelit – përdoruesi kalon në një hap më të avancuar.
Besimi ≤ 2	Përsëritje – përdoruesi e përsërit detyrën me një kërkesë më të thjeshtë dhe më të qartë.
Në rastet e tjera	Vazhdim – përdoruesi vazhdon me hapin e radhës, me një kërkesë më të fokusuar.

Po ashtu, `nextAction` përshtatet: për besim të ulët (≤2) sugjeron të ripunohet vetëm pjesa më e dobët; për besim të lartë (≥4) ngre shiritin duke kërkuar përmirësimin e detajit që një rishikues do ta vinte në dyshim; ndryshe sugjeron përmirësimin e pjesës më të dobët në misionin pasues.

Kjo logjikë e mban përdoruesin brenda zonës së zhvillimit proksimal (Vygotsky, 1978): as të mbingarkuar, as të mërzitur.

8.5 Diagrami i vendimit adaptiv

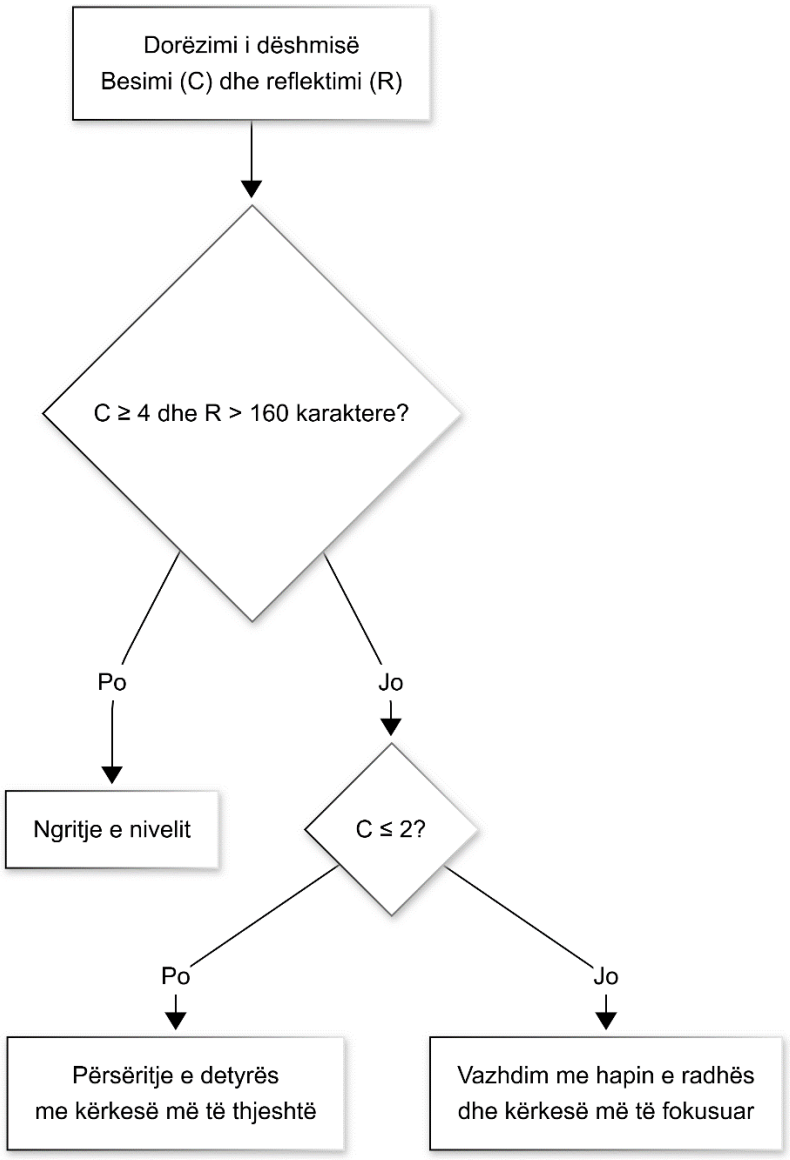

Figura 9. *Diagrami i mekanizmit të përshtatjes adaptive.*

8.6 Progresi i XP-së, niveleve dhe aftësive

Pas dorëzimit të parë të një misioni, sistemi përditëson automatikisht progresin e nxënësit. Shpërblimi me XP shpërndahet ndërmjet aftësive që lidhen me misionin, ndërsa niveli i secilës aftësi rritet në varësi të XP-së së grumbulluar. Në të njëjtën kohë, përditësohen edhe XP-ja dhe niveli i përgjithshëm i përdoruesit.

Po ashtu, sistemi:

- rrit `streak`-un me 1 për çdo dorëzim të ri;
- jep një arritje sipas llojit të misionit, si `final-artifact` për `FINAL_BOSS`, `mini-boss-clear` për `MINI_BOSS` dhe `first-proof` për misionet e tjera;
- siguron që shpërblimet të jepen vetëm gjatë dorëzimit të parë të misionit.

Kjo e fundit parandalon që përdoruesi të fitojë XP ose arritje shtesë përmes ridorëzimeve të njëpasnjëshme dhe ndihmon që progresi i regjistruar në sistem të pasqyrojë punën e kryer realisht.

8.7 Harta e misioneve

Harta e rrugës paraqet në mënyrë të qartë strukturën dhe progresin e nxënësit, duke dalluar misionet e përfunduara, misionin aktiv, pikat e kontrollit, nyjën e përshtatjes dhe misionet e kyçura. Gjendja e secilit mision bazohet në të dhënat reale të dorëzimeve dhe pasqyron progresin aktual të nxënësit.

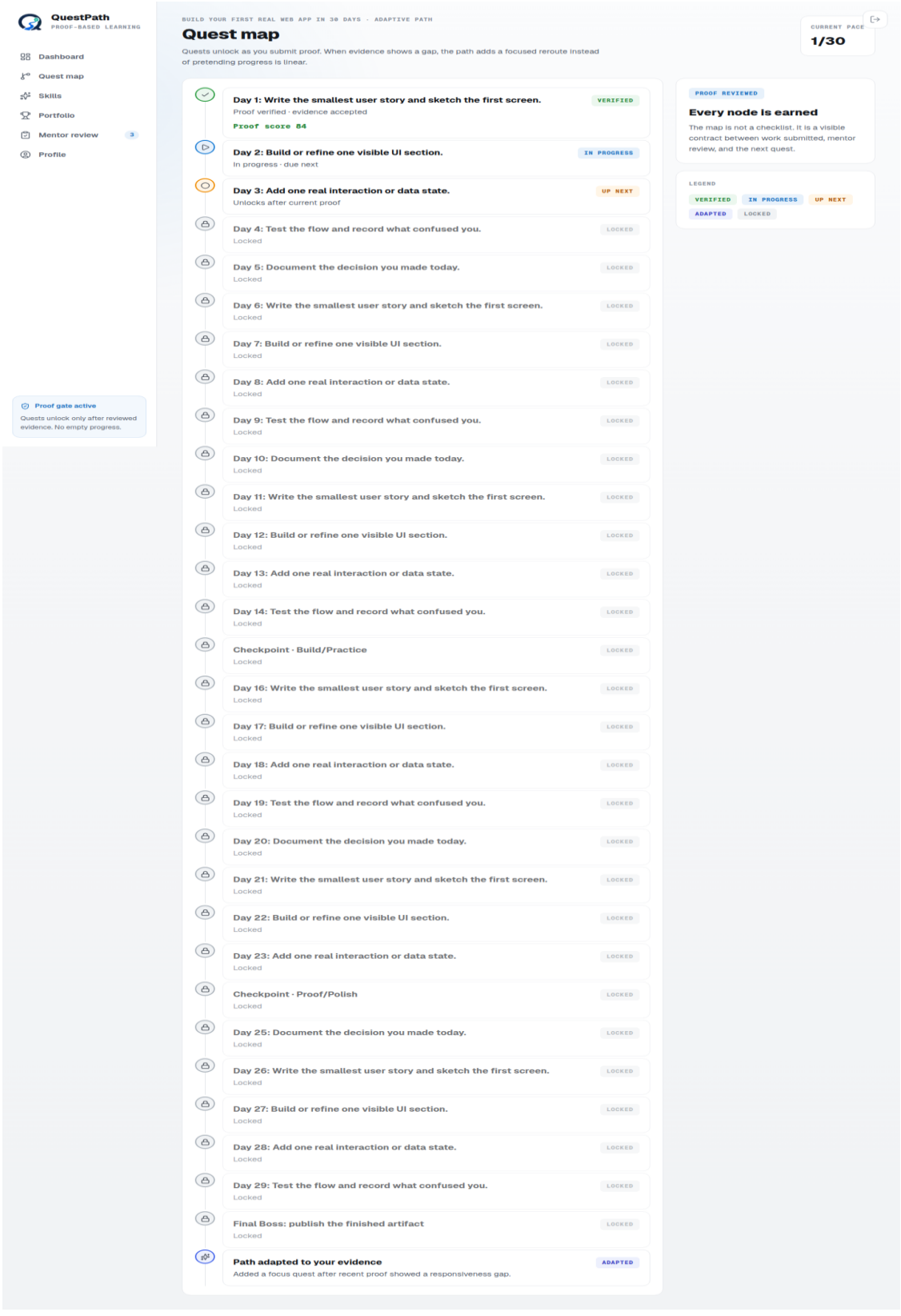

Figura 4. *Harta e misioneve - pamja në pajisje mobile.*

8.8 Ndërfaqja e mentorit dhe roli i tij në vlerësim

Përveç reagimit automatik, QuestPath mundëson edhe vlerësimin nga mentori përmes një hapësire të veçantë të sistemit. Mentori mund të shqyrtojë dëshmitë e dorëzuara, t'i vlerësojë ato sipas kritereve si qartësia, struktura, mënyra e prezantimit dhe reflektimi, si dhe të japë komente të shpejta. Gjithashtu, ai mund të shohë se si ky vlerësim ndikon në përshtatjen e misionit të ardhshëm.

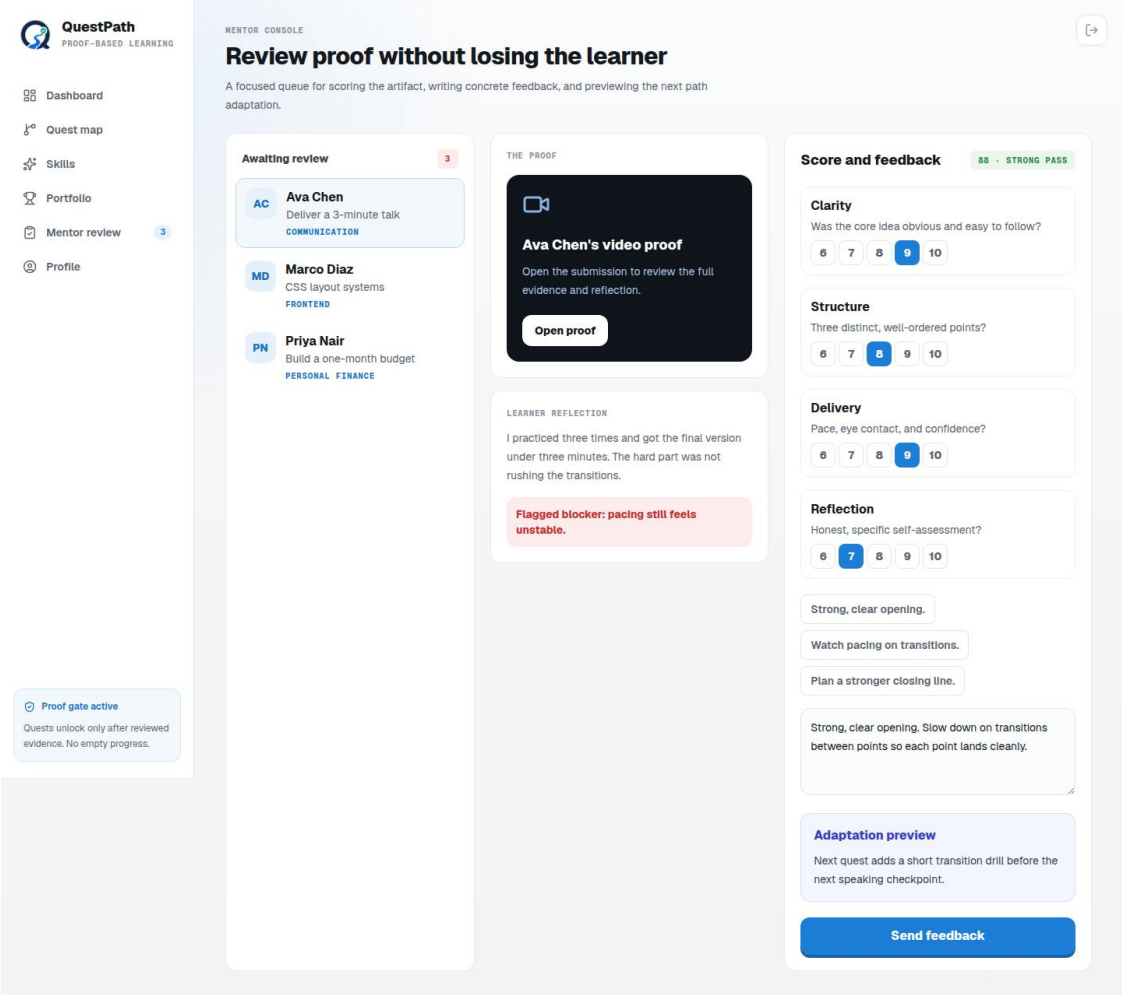

Figura 5. *Ndërfaqja e mentorit për shqyrtimin e punës.*

Në versionin aktual të prototipit, pjesa për rishikimin nga mentori është realizuar për të demonstruar mënyrën e funksionimit të procesit të vlerësimit, por ende nuk është e lidhur plotësisht me të dhënat reale të sistemit. Të dhënat e paraqitura janë të parapërgatitura, ndërsa vlerësimet dhe komentet e mentorit nuk ruhen ende në bazën e të dhënave. Deri në integrimin e plotë të kësaj pjese, reagimi automatik përdoret për dorëzimet reale të përdoruesve.

8.9 Qasja e bazuar në rregulla dhe kufizimet e saj

Përdorimi i një qasjeje të bazuar në rregulla, në vend të një modeli të mësimit makinerik, është bërë me qëllim. Kjo qasje është zgjedhur për disa arsye kryesore:

- **Qartësia:** përdoruesi dhe mentori mund ta kuptojnë lehtë pse sistemi ka marrë një vendim të caktuar ose pse është ndryshuar rruga mësimore.
- **Sasia e të dhënave:** prototipi aktual nuk disponon ende të dhëna të mjaftueshme për trajnimin dhe vlerësimin e një modeli të mësimit makinerik.
- **Besueshmëria:** rregullat e përcaktuara paraprakisht janë më të lehta për t'u testuar dhe japin rezultate të parashikueshme.
- **Aspekti etik:** kjo qasje zvogëlon rrezikun e vendimeve të paqarta ose të ndikuara nga paragjykime të fshehura, gjë që është veçanërisht e rëndësishme në një sistem arsimor.

Kjo zgjedhje nuk e përjashton përdorimin e mësimit makinerik në të ardhmen. Përkundrazi, qasja aktuale krijon një bazë të qartë krahasimi për zgjidhje më të avancuara që mund të zhvillohen në fazat e ardhshme të sistemit.

9 TESTIMI DHE VERIFIKIMI

9.1 Strategjia e verifikimit

Siç u theksua në metodologji, ky punim nuk përfshin një studim empirik me përdorues. Verifikimi është, pra, *i bazuar në prototip dhe skenarë*: ai konfirmon korrektësinë funksionale, koherencën e rrjedhës dhe qëndrueshmërinë operacionale, por jo efektivitetin pedagogjik. Të gjitha rezultatet e raportuara këtu rrjedhin nga regjistrimet e verifikimit të prototipit dhe nuk janë të sajuara.

9.2 Kontrollet statike të kodit

Për të verifikuar cilësinë dhe korrektësinë e kodit janë kryer disa kontrolle teknike:

- **ESLint** (`pnpm run lint`) është ekzekutuar pa gabime, duke treguar se kodi respekton rregullat e përcaktuara për stilin dhe cilësinë.
- **Ndërtimi i aplikacionit** (`pnpm run build`) ka përfunduar me sukses, duke konfirmuar se aplikacioni mund të ndërtohet dhe ekzekutohet pa gabime gjatë procesit të përgatitjes për publikim.
- **`git diff --check`** është ekzekutuar pa probleme, duke verifikuar se nuk ka gabime të formatimit ose shenja të mbetura nga konfliktet në kod.

Përdorimi i TypeScript (TypeScript, 2026) dhe modeleve të përcaktuara me *Prisma* ofron gjithashtu një nivel shtesë kontrolli, pasi shumë gabime mund të zbulohen që në fazën e zhvillimit, para se aplikacioni të ekzekutohet.

9.3 Testimi i funksionaliteteve kryesore

Për të verifikuar qasjen në faqet kryesore të aplikacionit, janë kryer teste si për përdoruesit e paautentikuar, ashtu edhe për ata të autentikuar.

- **Pa autentikim:** faqja `/login` hapet normalisht, ndërsa `/` dhe `/dashboard` e ridrejtojnë përdoruesin te faqja e hyrjes.
- **Me autentikim:** faqet `/dashboard`, `/map`, `/quests/[id]`, `/submit/[id]`, `/feedback/[id]`, `/portfolio`, `/skills`, `/profile` dhe `/review` hapen me sukses për një përdorues me sesion aktiv.

Këto teste tregojnë se qasja në faqet e mbrojtura funksionon siç është paraparë: përdoruesit pa sesion ridrejtohen te faqja e hyrjes, ndërsa përdoruesit e autentikuar mund t'i hapin normalisht faqet përkatëse.

9.4 Testimi i ndërfaqes në pajisje mobile

Gjatë testimit në pajisje mobile u vërejt se disa pjesë të panelit kryesor tejkalonin gjerësinë e ekranit dhe shkaktonin lëvizje horizontale. Për këtë arsye, u bënë disa rregullime në paraqitjen e aplikacionit, duke përshtatur panelet, kartat, navigimin mobil dhe seksionet e dashboard-it për ekrane më të vogla. Pas këtyre ndryshimeve, aplikacioni u testua me Playwright dhe u konfirmua se faqet kryesore shfaqeshin pa lëvizje horizontale.

9.5 Verifikimi i publikimit të aplikacionit

Vendosja e aplikacionit në produksion u verifikua përmes Coolify-t dhe kontrolleve të drejtpërdrejta. Aplikacioni i vendosur në server rezultoi funksional dhe në gjendje të rregullt. Faqja e hyrjes dhe faqet kryesore u hapën siç pritej, ndërsa përdoruesit e paautentikuar ridrejtohen te faqja e hyrjes. Përdoruesit me sesion aktiv mund të qasen normalisht në pjesët kryesore të sistemit. Gjithashtu, gjatë nisjes së aplikacionit nuk u evidentuan gabime dhe testet në pajisje mobile konfirmuan se faqet shfaqen pa lëvizje horizontale të panevojshme.

9.6 Përmbledhja e rezultateve

Kategoria	Kontrolli	Rezultati
Kontrolli i kodit	*`pnpm run lint`*	Kaloi me sukses
Kontrolli i kodit	*`pnpm run build`*	Kaloi me sukses
Kontrolli i kodit	*`git diff --check`*	Kaloi me sukses
Testimi lokal	*`/login`* i paautentikuar	U hap me sukses
Testimi lokal	*/*, */dashboard* të paautentikuara	Ridrejtim te `/login`
Testimi lokal	Rrugët e autentikuara	U hap me sukses
Testimi në pajisje mobile	Kontrolli i paraqitjes në ekrane me gjerësi të ndryshme	Kaloi
Vendosja në server	Gjendja e aplikacionit në Coolify	Funksional
Testimi në produksion	Hapja e faqes `/login`	U hap me sukses

Kategoria	Kontrolli	Rezultati
Testimi në produksion	Qasja në faqet e mbrojtura me autentikim	U hap me sukses
Testimi në produksion	Paraqitja në pajisje mobile	Pa lëvizje horizontale

Tabela 12. *Rezultatet e testimit dhe verifikimit.*

9.7 Kufizimet e testimit

Gjatë vlerësimit të prototipit duhet të merren parasysh disa kufizime:

- **Mungesa e testimit me përdorues:** në këtë fazë nuk është matur ndikimi i sistemit në rezultatet e të nxënit, angazhimin apo kënaqësinë e përdoruesve. Për këtë arsye, aspektet pedagogjike mbeten për t'u vlerësuar në studime të ardhshme.
- **Mungesa e testeve të automatizuara në nivel njësie dhe integrimi:** verifikimi është mbështetur kryesisht në kontrolle funksionale dhe teste përmes shfletuesit. Zhvillimi i një grupi më të plotë testesh automatike mbetet një drejtim për përmirësim të mëtejshëm.
- **Pjesa e mentorit përdor të dhëna shembull:** funksionimi i ndërfaqes së mentorit është verifikuar, por ende nuk është testuar plotësisht me dorëzime reale të ruajtura në bazën e të dhënave.
- **Ndryshim në pamjen e faqes së hyrjes:** pamja e paraqitur në një nga figurat i përket një versioni vizual më të hershëm të aplikacionit. Ky ndryshim është vetëm vizual dhe nuk ndikon në funksionimin e procesit të hyrjes.

Këto kufizime nuk e pengojnë vlerësimin e funksionimit teknik të prototipit, por tregojnë qartë se cilat pjesë janë verifikuar dhe cilat kërkojnë testim dhe vlerësim të mëtejshëm.

10 SIGURIA DHE PRIVATËSIA

Ky kapitull analizon qëndrimin e sigurisë së prototipit në mënyrë të ndershme, duke dalluar masat e implementuara nga kufizimet e njohura. Analiza strukturohet pjesërisht sipas OWASP Top 10 (OWASP Foundation, 2021).

10.1 Siguria e autentikimit

Fjalëkalimet nuk ruhen në formë të lexueshme, por mbrohen duke përdorur algoritmin `bcrypt` me faktor kostoje 12 (Provos & Mazières, 1999). Kjo mënyrë e ruajtjes e vështirëson ndjeshëm provimin e një numri të madh kombinimesh të fjalëkalimeve dhe rrit sigurinë e llogarive. Gjatë procesit të hyrjes, fjalëkalimi i dhënë nga përdoruesi verifikohet përmes funksionit `bcrypt.compare`.

Sistemi gjithashtu i trajton në mënyrë të njëjtë rastet kur email-i nuk ekziston dhe kur fjalëkalimi është i pasaktë. Në të dyja rastet shfaqet një mesazh i përgjithshëm gabimi, pa treguar nëse një adresë email-i është e regjistruar apo jo. Kjo ndihmon në mbrojtjen e informacionit rreth llogarive të përdoruesve.

10.2 Integriteti i sesionit

Sesioni i përdoruesit ruhet përmes një cookie-je të nënshkruar, e cila lidhet me identifikuesin e përdoruesit dhe mbrohet me një çelës sekret në server. Për të rritur sigurinë, cookie-ja përdor disa masa mbrojtëse:

- `httpOnly` – pengon qasjen në cookie përmes JavaScript-it në shfletues;
- `sameSite: "lax"` – ndihmon në mbrojtjen ndaj kërkesave të padëshiruara nga faqe të tjera;
- `secure` – në produksion, cookie-ja dërgohet vetëm përmes lidhjeve HTTPS;
- `maxAge` prej 30 ditësh – përcakton kohëzgjatjen e vlefshmërisë së sesionit.

Gjatë çdo kërkese, sistemi kontrollon vlefshmërinë e cookie-së për t'u siguruar që të dhënat e sesionit nuk janë ndryshuar ose manipuluar.

10.3 Kontrolli i qasjes në të dhëna

Çdo faqe dhe veprim që kërkon autentikim kontrollon nëse përdoruesi ka një sesion të vlefshëm. Nëse sesioni mungon ose nuk është i vlefshëm, përdoruesi ridrejtohet te faqja e hyrjes. Përveç kësaj, sistemi kontrollon nëse përdoruesi ka të drejtë të hapë një mision të caktuar. Një mision mund të hapet vetëm nëse ai i përket rrugës mësimore të përdoruesit. Kjo ndihmon në parandalimin e qasjes së paautorizuar në të dhëna ose përmbajtje që nuk i përkasin përdoruesit.

10.4 Menaxhimi i sekreteve dhe izolimi

Të dhënat e ndjeshme, si adresa e bazës së të dhënave, çelësi i autentikimit dhe kredencialet e përdoruesit demonstrues, ruhen veçmas nga kodi i aplikacionit. Në mjedisin lokal ato ruhen në .env.local, ndërsa në produksion menaxhohen përmes konfigurimit të Coolify-t. Këto të dhëna nuk ruhen në historikun e kodit.

Gjithashtu, skedari .dockerignore pengon përfshirjen e skedarëve lokalë dhe të panevojshëm gjatë krijimit të aplikacionit për server. Baza e të dhënave funksionon e ndarë nga pjesët e tjera të sistemit, duke zvogëluar mundësinë e qasjes së panevojshme nga shërbimet e tjera.

10.5 Analiza sipas OWASP Top 10 (2021)

Kategoria OWASP	Gjendja në QuestPath
A01 – Kontrolli i qasjes	Është trajtuar përmes kontrollit të sesionit dhe verifikimit nëse përdoruesi ka të drejtë të hapë të dhënat ose misionet përkatëse.
A02 – Mbrojtja e të dhënave	Fjalëkalimet mbrohen me `bcrypt`, ndërsa sesionet ruhen dhe verifikohen në mënyrë të sigurt.
A03 – Futja e kodit të dëmshëm	Rreziku zvogëlohet përmes Prisma-s, e cila menaxhon kërkesat ndaj bazës së të dhënave pa përdorur drejtpërdrejt SQL të papërpunuar.
A04 – Dizajni i pasigurt	Është trajtuar pjesërisht; sistemi përfshin masa mbrojtëse, por disa pjesë të prototipit kanë ende kufizime.
A05 – Konfigurimi i pasigurt	Të dhënat e ndjeshme ruhen jashtë kodit dhe konfigurimi i mjedisit të produksionit është ndarë nga ai lokal.
A06 – Komponentë të vjetruar ose të cenueshëm	Varësitë e projektit menaxhohen me versione të përcaktuara dhe një skedar të pandryshueshëm të varësive.

Kategoria OWASP	Gjendja në QuestPath
A07 – Probleme me identifikimin dhe autentikimin	Fjalëkalimet mbrohen në mënyrë të sigurt, por prototipi nuk përfshin ende kufizim të numrit të tentimeve apo autentikim me më shumë se një faktor.
A08 – Integriteti i softuerit dhe të dhënave	Varësitë instalohen sipas versioneve të përcaktuara në projekt, duke zvogëluar rrezikun e ndryshimeve të paqëllimshme.
A09 – Regjistrimi dhe monitorimi	Monitorimi është i kufizuar dhe përfshin kryesisht kontrollin e gjendjes së aplikacionit dhe regjistrimet gjatë nisjes.
A10 – Kërkesa të paautorizuara nga serveri	Rreziku është i ulët, pasi lidhjet e dëshmive ruhen në sistem, por nuk hapen apo shkarkohen automatikisht nga serveri.

Tabela 13. *Analiza e sigurisë sipas OWASP Top 10 (2021).*

10.6 Konsideratat e privatësisë

Të dhënat e përdorura aktualisht në sistem janë tërësisht shembull dhe nuk përfaqësojnë përdorues realë. Megjithatë, dizajni i sistemit merr parasysh disa parime të privatësisë për përdorim të ardhshëm, si ruajtja vetëm e të dhënave të nevojshme, mbrojtja e fjalëkalimeve dhe mundësia për të kontrolluar dukshmërinë publike të profilit dhe portofolit. Për një përdorim real të sistemit, do të nevojiteshin edhe politika të qarta për ruajtjen, fshirjen dhe pëlqimin për përpunimin e të dhënave.

10.7 Kufizimet e njohura të sigurisë

Sistemi aktual i sesionit është ndërtuar për qëllime prototipi dhe ka disa kufizime:

1. Sesioni nuk përmban një afat skadimi brenda vetes; kohëzgjatja e tij përcaktohet vetëm nga cookie-s.
2. Nuk ekziston ende mundësia për të anuluar sesionet aktive, për shembull pas ndryshimit të fjalëkalimit.
3. Nuk ka kufizim të numrit të tentimeve për hyrje, prandaj mbrojtja ndaj provave të shumta të fjalëkalimit është e kufizuar.
4. Sistemi nuk përdor autentikim me dy ose më shumë hapa.

Këto kufizime janë të pranueshme për një prototip, por do të duhej të përmirësoheshin para përdorimit të sistemit me përdorues realë. Përmirësimi i sigurisë së sesionit mbetet pjesë e punës së ardhshme.

11 KUFIZIMET DHE PUNA E ARDHSHME

11.1 Kufizimet

Tabela e mëposhtme paraqet kufizimet kryesore të prototipit, të ndara sipas kategorive. Këto kufizime janë marrë parasysh gjatë zhvillimit dhe tregojnë pjesët që kërkojnë përmirësim ose zhvillim të mëtejshëm.

Lloji	Kufizimi	Ndikimi
Vlerësimi	Nuk është zhvilluar studim me përdorues realë	Efektiviteti pedagogjik i sistemit ende nuk është vërtetuar në praktikë
Përshtatja	Vendimet e sistemit bazohen në rregulla të përcaktuara paraprakisht	Qasja është e thjeshtë dhe e kuptueshme, por mund të përmirësohet më tej
Përshtatja	Reflektimi vlerësohet kryesisht sipas gjatësisë së tekstit	Gjatësia nuk e pasqyron gjithmonë cilësinë reale të reflektimit
Mentori	Pjesa e mentorit përdor të dhëna shembull	Funksionimi i ndërfaqes është demonstruar, por ende nuk është lidhur plotësisht me bazën e të dhënave
Testimi	Mungojnë testet e plota automatike për pjesët individuale dhe ndërveprimin mes tyre	Testimi është mbështetur kryesisht në kontrolle funksionale dhe përmes shfletuesit
Siguria	Menaxhimi i sesionit ka ende disa kufizime	Nevojiten masa shtesë sigurie para përdorimit me përdorues realë
Funksionaliteti	Pagesat dhe abonimet nuk janë implementuar plotësisht	Sistemi ruan vetëm interesimin për një plan abonimi, pa proces real pagese
Të dhënat	Ndërfaqja është ndërtuar kryesisht për një rrugë aktive në të njëjtën kohë	Modeli i të dhënave mund të mbështesë më shumë rrugë, por ndërfaqja kërkon përshtatje

11.2 Puna e ardhshme

11.2.1 Vlerësimi empirik

Një nga hapat më të rëndësishëm për zhvillimin e mëtejshëm të QuestPath është testimi i tij me përdorues realë. Në një studim të ardhshëm, sistemi mund të krahasohet me një mënyrë më tradicionale të organizimit të detyrave, duke vlerësuar përfundimin e aktiviteteve, cilësinë e

dëshmive të dorëzuara, angazhimin dhe perceptimin e studentëve për progresin e tyre. Studimi mund të përfshijë gjithashtu intervista me 10–20 studentë pas një periudhe përdorimi prej 14 deri në 30 ditë. Pyetja kryesore do të ishte nëse përshtatja e rrugës mësimore në bazë të dëshmive dhe reagimit e ndihmon studentin të mësojë në mënyrë më të dobishme dhe më të menaxhueshme. Një studim i tillë do të mundësonte vlerësimin e efektivitetit të sistemit në praktikë dhe do të siguronte rezultate konkrete për përmirësimin e tij.

11.2.2 Integrimi i plotë i rishikimit njerëzor

Në të ardhmen, sistemi mund të përmirësohet duke marrë parasysh edhe tregues të tjerë, si cilësia e reflektimit dhe rezultatet e mëparshme të nxënësit. Me grumbullimin e më shumë të dhënave, mund të shqyrtohet edhe përdorimi i modeleve më të avancuara për ndjekjen e progresit dhe përshtatjen e rrugës mësimore. Megjithatë, çdo qasje e tillë duhet të mbetet e kuptueshme dhe të mundësojë shpjegimin e vendimeve që merr sistemi.

11.2.3 Përmirësimi i mekanizmit adaptiv

Në të ardhmen, sistemi mund të përmirësohet duke marrë parasysh edhe tregues të tjerë, si cilësia e reflektimit dhe rezultatet e mëparshme të nxënësit. Me grumbullimin e më shumë të dhënave, mund të shqyrtohet edhe përdorimi i modeleve më të avancuara për ndjekjen e progresit dhe përshtatjen e rrugës mësimore. Megjithatë, çdo qasje e tillë duhet të mbetet e kuptueshme dhe të mundësojë shpjegimin e vendimeve që merr sistemi.

11.2.4 Përmirësimi i cilësisë dhe besueshmërisë

Besueshmëria e sistemit mund të përmirësohet përmes testeve automatike, kontrollit të vazhdueshëm të kodit dhe monitorimit më të mirë të funksionimit të aplikacionit.

11.2.5 Përmirësimi i sigurisë

Siguria mund të përmirësohet përmes sesioneve me afat të përcaktuar, mundësisë së anulimit të tyre, kufizimit të tentimeve për hyrje dhe, sipas nevojës, autentikimit me më shumë se një hap. Gjithashtu, duhet të përcaktohen politika të qarta për ruajtjen dhe fshirjen e të dhënave të përdoruesve.

11.2.6 Veçori produkti

Në të ardhmen, sistemi mund të zgjerohet me pagesa dhe abonime të plota, mundësi për përdorimin e disa rrugëve mësimore njëkohësisht, si dhe ndarje publike të portofolit me kontroll

mbi privatësinë. Gjithashtu, mund të bëhen përmirësime të mëtejshme në qasshmëri, duke u bazuar në standardet WCAG 2.1 (W3C, 2018).

12 PËRFUNDIM

Ky punim paraqiti zhvillimin e *QuestPath*, një platformë për mësim të vetëdrejtuar, ku progresi i nxënësit lidhet me punën konkrete që ai realizon dhe dëshmon gjatë procesit të të nxënit. Qëllimi kryesor ishte të krijohej një sistem ku përparimi nuk mbështetet vetëm në përfundimin e aktiviteteve, por në dorëzimin e dëshmive, reflektimin mbi punën e kryer dhe reagimin që nxënësi merr pas çdo misioni.

Në përgjigje të pyetjeve kërkimore, sistemi u ndërtua rreth ciklit *"mision → dëshmi → reagim → përshtatje"*, duke e vendosur dëshminë në qendër të progresit të nxënësit. Përshtatja e hapit të ardhshëm bazohet në dy tregues kryesorë: nivelin e besimit të nxënësit dhe reflektimin e tij mbi punën e realizuar. Për realizimin e sistemit u përdorën **Next.js, Prisma dhe PostgreSQL**, të cilat mundësuan ndërtimin e funksioneve kryesore dhe menaxhimin e të dhënave të aplikacionit. Funksionimi teknik i prototipit u verifikua përmes kontrolleve të kodit, testimit të faqeve kryesore, testimit në pajisje mobile dhe verifikimit të funksionimit të aplikacionit pas vendosjes në server.

Kontributi kryesor i punimit qëndron në paraqitjen e një qasjeje ku nxënësi avancon përmes dëshmive konkrete të punës së tij. Pas çdo dorëzimi, ai merr reagim dhe udhëzime për hapin e ardhshëm, ndërsa rruga mësimore përshtatet sipas progresit të treguar. Organizimi i reagimit mbështetet në parimet e Hattie dhe Timperley, ndërsa mënyra e përshtatjes së vështirësisë lidhet me konceptin e zonës së zhvillimit proksimal. Në këtë mënyrë, QuestPath synon ta bëjë progresin më të dukshëm, të kuptueshëm dhe të lidhur me punën reale të nxënësit.

Megjithatë, sistemi i zhvilluar është ende një prototip dhe ka disa kufizime. Në kuadër të këtij punimi nuk është realizuar testim me përdorues realë, pjesa e mentorit përdor të dhëna shembull dhe disa mekanizma të sigurisë kërkojnë zhvillim të mëtejshëm. Për këtë arsye, një hap i rëndësishëm në të ardhmen është testimi i QuestPath me përdorues, për të vlerësuar jo vetëm funksionimin teknik, por edhe ndikimin e tij në angazhim, progres dhe procesin e të nxënit.

Në përfundim, QuestPath paraqet një mënyrë të strukturuar të organizimit të të nxënit, ku nxënësi përparon duke **realizuar, dëshmuar, reflektuar dhe përmirësuar** punën e tij. Prototipi i zhvilluar tregon se kjo qasje mund të realizohet në praktikë dhe krijon një bazë të mirë për zhvillimin dhe vlerësimin e mëtejshëm të sistemit.

13 LITERATURA

Burimet e përdorura në punim janë renditur alfabetikisht sipas autorit.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191–215.

Black, P., & Wiliam, D. (1998). Assessment and Classroom Learning. *Assessment in Education: Principles, Policy & Practice*, 5(1), 7–74.

Bloom, B. S. (1968). Learning for Mastery. *Evaluation Comment*, 1(2). UCLA Center for the Study of Evaluation.

Bloom, B. S. (1984). The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher*, 13(6), 4–16.

Coolify. (2026). *Coolify Documentation*. https://coolify.io/docs (qasur më 1.9.2026).

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. New York: Harper & Row.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining "Gamification". *Proceedings of the 15th International Academic MindTrek Conference*, 9–15.

Dewey, J. (1938). *Experience and Education*. New York: Macmillan / Kappa Delta Pi.

Docker Inc. (2026). *Docker Documentation — Multi-stage builds*. https://docs.docker.com/build/building/multi-stage/ (qasur më 1.9.2026).

Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review*, 100(3), 363–406.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation). University of California, Irvine.

Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Boston: Addison-Wesley.

Hattie, J., & Timperley, H. (2007). The Power of Feedback. *Review of Educational Research*, 77(1), 81–112.

Knowles, M. S. (1975). *Self-Directed Learning: A Guide for Learners and Teachers*. New York: Association Press.

Meta Open Source. (2026). *React Documentation*. https://react.dev (qasur më 1.9.2026).

Nielsen, J. (1994). *10 Usability Heuristics for User Interface Design*. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/ (qasur më 1.9.2026).

OWASP Foundation. (2021). *OWASP Top 10 — 2021*. https://owasp.org/Top10/ (qasur më 1.9.2026).

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

PostgreSQL Global Development Group. (2026). *PostgreSQL 16 Documentation*. https://www.postgresql.org/docs/ (qasur më 1.9.2026).

Prisma. (2026). *Prisma ORM Documentation*. https://www.prisma.io/docs (qasur më 1.9.2026).

Provos, N., & Mazières, D. (1999). A Future-Adaptable Password Scheme. *Proceedings of the USENIX Annual Technical Conference (FREENIX Track)*, 81–91.

Roediger, H. L., & Karpicke, J. D. (2006). Test-Enhanced Learning: Taking Memory Tests Improves Long-Term Retention. *Psychological Science*, 17(3), 249–255.

Ryan, R. M., & Deci, E. L. (2000). Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being. *American Psychologist*, 55(1), 68–78.

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science*, 12(2), 257–285.

Tailwind Labs. (2026). *Tailwind CSS Documentation*. https://tailwindcss.com/docs (qasur më 1.9.2026).

TypeScript. (2026). *TypeScript Handbook*. https://www.typescriptlang.org/docs/ (qasur më 1.9.2026).

Vercel. (2026). *Next.js Documentation — App Router*. https://nextjs.org/docs (qasur më 1.9.2026).

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press.

World Wide Web Consortium (W3C). (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. https://www.w3.org/TR/WCAG21/ (qasur më 1.9.2026).

Zubizarreta, J. (2009). *The Learning Portfolio: Reflective Practice for Improving Student Learning* (2nd ed.). San Francisco: Jossey-Bass.

14 SHTOJCAT

14.1 Shtojca A — Skema e plotë e të dhënave (Prisma)

Në këtë shtojcë paraqitet skema e plotë e bazës së të dhënave të QuestPath, e përcaktuar përm es Prisma.

```
enum QuestState { LOCKED  AVAILABLE  COMPLETED  REPEATED }
enum QuestKind  { STANDARD  MINI_BOSS  FINAL_BOSS }
enum SubscriptionPlan { FREE  PRO_PREVIEW }

model User {
  id            String   @id @default(cuid())
  email         String   @unique
  name          String
  passwordHash  String
  xp            Int      @default(0)
  level         Int      @default(1)
  streak        Int      @default(0)
  plan          SubscriptionPlan @default(FREE)
  createdAt     DateTime @default(now())
  updatedAt     DateTime @updatedAt
  paths         UserPath[]
  submissions   QuestSubmission[]
  skillProgress UserSkillProgress[]
  achievements  UserAchievement[]
}

model PathTemplate {
  id          String  @id @default(cuid())
  slug        String  @unique
  title       String
  duration    Int
  description String
  artifact    String
  quests      Quest[]
  userPaths   UserPath[]
  createdAt   DateTime @default(now())
```

```
}

model UserPath {
  id             String   @id @default(cuid())
  userId         String
  templateId     String
  active         Boolean  @default(true)
  currentDay     Int      @default(1)
  currentLevel   String
  minutesPerDay  Int
  difficulty     String
  deadline       DateTime?
  usualBlocker   String
  completedQuests Int     @default(0)
  createdAt      DateTime @default(now())
  updatedAt      DateTime @updatedAt
  user           User        @relation(fields: [userId], references: [id], onDelete: Cascade)
  template       PathTemplate @relation(fields: [templateId], references: [id])
  @@index([userId, active])
}

model Quest {
  id           String   @id @default(cuid())
  templateId   String
  day          Int
  stage        String
  title        String
  mission      String
  why          String
  timeEstimate  Int
  proofRequired String
  reflection   String
  xpReward     Int
  kind         QuestKind @default(STANDARD)
  skillKeys    String[]
  template     PathTemplate @relation(fields: [templateId], references: [id], onDelete: Cascade)
  submissions  QuestSubmission[]
```

```
  @@unique([templateId, day])
  @@index([templateId, day])
}

model QuestSubmission {
  id         String   @id @default(cuid())
  userId     String
  questId    String
  reflection String
  proofUrl   String?
  notes      String?
  createdAt  DateTime @default(now())
  user       User     @relation(fields: [userId], references: [id], onDelete: Cascade)
  quest      Quest    @relation(fields: [questId], references: [id])
  feedback   Feedback?
  @@unique([userId, questId])
  @@index([userId, createdAt])
}

model Feedback {
  id                String @id @default(cuid())
  submissionId      String @unique
  wentWell          String
  missing           String
  improvement       String
  nextAction        String
  adaptationDecision String
  createdAt         DateTime @default(now())
  submission        QuestSubmission @relation(fields: [submissionId], references: [id], onDelete: Casca
de)
}

model Skill {
  id          String @id @default(cuid())
  key         String @unique
  name        String
  description String
```

```
  progress    UserSkillProgress[]
}

model UserSkillProgress {
  id        String   @id @default(cuid())
  userId    String
  skillId   String
  xp        Int      @default(0)
  level     Int      @default(1)
  updatedAt DateTime @updatedAt
  user      User  @relation(fields: [userId], references: [id], onDelete: Cascade)
  skill     Skill @relation(fields: [skillId], references: [id])
  @@unique([userId, skillId])
}

model Achievement {
  id          String @id @default(cuid())
  key         String @unique
  name        String
  description String
  icon        String
  users       UserAchievement[]
}

model UserAchievement {
  id           String   @id @default(cuid())
  userId       String
  achievementId String
  earnedAt     DateTime @default(now())
  user         User       @relation(fields: [userId], references: [id], onDelete: Cascade)
  achievement   Achievement @relation(fields: [achievementId], references: [id])
  @@unique([userId, achievementId])
}

model SubscriptionIntent {
  id        String @id @default(cuid())
  userId    String
```

```
  plan      SubscriptionPlan
  reason    String
  createdAt DateTime @default(now())
}
```

14.2 Shtojca B — Logjika e gjenerimit të misioneve

Në vijim paraqiten funksionet kryesore të përdorura për përcaktimin e fazës dhe llojit të misionit, në varësi të ditës dhe kohëzgjatjes së rrugës mësimore:

```
function stageFor(day: number, duration: number) {
  const pct = day / duration;
  if (pct <= 0.25) return "Foundation";
  if (pct <= 0.65) return "Build/Practice";
  if (pct < 1)     return "Proof/Polish";
  return "Final Boss";
}

function questKind(day: number, duration: number) {
  if (day === duration) return QuestKind.FINAL_BOSS;
  if (day === Math.ceil(duration / 2) || day === Math.ceil(duration * 0.8))
    return QuestKind.MINI_BOSS;
  return QuestKind.STANDARD;
}
```

Në sistem janë përcaktuar aftësitë planning, html-css, javascript, react, ui-polish, deployment d he communication. Gjithashtu, janë përfshirë arritjet first-proof, mini-boss-clear dhe final-artif act. Sistemi përmban tri rrugë mësimore: first-real-web-app me kohëzgjatje 30 ditë, junior-dev eloper-ready me 30 ditë dhe thesis-presentation me 14 ditë.

14.3 Shtojca C — Rezultatet e verifikimit të prototipit

Gjatë zhvillimit të QuestPath janë kryer disa kontrolle për të verifikuar funksionimin e aplikacionit dhe vendosjen e tij në server. Projekti është menaxhuar përmes *Git* dhe është vendosur në produksion përmes *Coolify*.

Aplikacioni është publikuar *online*, ndërsa funksionimi i tij është kontrolluar si në *mjedisin lokal*, ashtu edhe pas vendosjes në *server*. Kontrollet kanë përfshirë ndërtimin e aplikacionit, kontrollin e kodit, qasjen në faqet kryesore, autentikimin dhe paraqitjen në pajisje mobile.

14.4 Shtojca D — Konfigurimi i mjedisit të aplikacionit

Aplikacioni përdor disa variabla të mjedisit për konfigurimin e bazës së të dhënave, autentikim it dhe llogarisë demonstrative. Vlerat reale të këtyre variablave ruhen vetëm në mjedisin lokal dhe në konfigurimin e Coolify-t dhe nuk përfshihen në kodin burimor.

Variablat kryesore janë:

```
DATABASE_URL
AUTH_SECRET
NEXTAUTH_SECRET
DEMO_EMAIL
DEMO_PASSWORD
```