

# Navodila za projekt in seminar

Računalniške storitve v oblaku (2025/2026)

## 1. Uvod

Cilj vaj pri predmetu Računalniške storitve v oblaku (RSO) je uspešna realizacija izbranega projekta spletne in/ali mobilne aplikacije, pri čemer je potrebno vključiti koncepte, obravnavane na predavanjih. Projekt se razvija v skupinah z dvema ali tremi študenti (če želite, lahko delati tudi sami), končan in uspešno zagovorjen projekt pa je pogoj za pristop k izpitu. Vaje lahko opravite tudi s seminarjem.

## 2. Potek razvoja in mejniki

Razvoj projekta poteka v tedenskem tempu in sledi predavanjem – predlagamo sprotno delo. Med semestrom bodo pomembnejši mejniki, za katere bodo dodatna navodila in obrazci za oddajo objavljena na spletni učilnici:

- 1. Razdelitev v skupine (rok: 19. 10. 2025):** Prijava skupine na spletni učilnici.
- 2. Prijava teme projekta ali seminarja (rok: 26. 10. 2025):** Oddaja prijave na spletni učilnici, prijava teme vključuje podroben opis izbrane teme, seznam članov ekipe, arhitekturno zasnovo rešitve, izbrano ogrodje ter povezavo do ustvarjene organizacije na GitHub oz. GitLab.
- 3. Neobvezna vmesna predstavitev (8.-14. 12. 2025):** Kratek javni »pitch« projekta ali seminarja na vajah, predstavitev arhitekturne zasnove in že razvitih delov.  
*[do 10 % ocene vaj]*
- 4. Oddaja projekta ali seminarja (rok: 11. 1. 2026):** Oddaja kode in poročila na spletni učilnici.  
*[do 10 % ocene vaj]*
- 5. Zagovor projekta ali seminarja (5.-18. 1. 2026):** Pregled bistvenih funkcionalnosti razvite aplikacije in vključenih konceptov iz seznama zahtev.  
*[do 80 % ocene vaj]*

### 3. Priprava na delo in razvojno okolje

Zahteva za uspešno opravljene vaje je, da vaš projekt sproti objavljate na GitHub (oziroma primerljivo platformo, na primer GitLab). Ustvarite si [organizacijo](#) v kateri boste ustvarili repozitorije za vašo aplikacijo. Repozitoriji naj bodo javni.

Na predavanjih bo uporabljeno mikrostoritveno ogrodje [KumuluzEE](#), ki nudi močno podporo za koncepte, obravnavane pri predmetu. Pri izdelavi projektov lahko uporabite katerikoli programski jezik in ogrodje, prav tako lahko različne mikrostoritve razvijate v različnih ogrodjih in programskih jezikih. Na spletni učilnici bo na voljo vzorčni skeletni primer za Javo in KumuluzEE.

Kot študenti Univerze v Ljubljani imate dostop do študentske licence za produkte in razvojna okolja JetBrains. Študentski račun is lahko ustvarite na [tem naslovu](#). Razmeroma popularno razvojno okolje je [Visual Studio Code](#). Na [GitHub Education](#) si dodatno lahko aktivirate študentski paket, v katerem se nahaja nekaj uporabnih orodij.

Pri razvoju svojih aplikacij boste lahko poskrbeli tudi za njihovo namestitev v oblak. V ta namen si ustvarite račun pri poljubnem ponudniku oblačnih storitev. Priporočamo, da izberete ponudnika, ki že ponuja prednameščeno okolje Kubernetes, saj vam bo to olajšalo delo. Nekaj možnosti:

- **Azure (Azure for Education):** \$100 kreditov. Za registracijo pojdite na [Azure for Students](#) in sledite postopku.
- **Google Cloud Platform:** Študenti lahko izkoristite brezplačen kredit.
- **AWS (Amazon Web Services):** AWS ponuja brezplačen nivo storitev za nove uporabnike.
- **DigitalOcean:** Ponuja enostavne oblačne storitve z opcijo upravljanja Kubernetes okolij.
- **IBM Cloud:** Omogoča uporabo brezplačnih storitev ter upravljanje Kubernetes klustrov.
- **Oracle Cloud Infrastructure:** Oracle ponuja brezplačni nivo storitev, ki vključuje dve vedno brezplačni virtualni računalniški instanci, baze podatkov, shranjevanje in več. Študenti lahko prav tako pridobijo dodatne kredite ob registraciji za izobraževalne namene.

Priporočamo, da za učinkovito organizacijo skupinskega dela ter sledenje nalogam in napredku projekta ustvarite GitHub projekt ali uporabite orodja, kot sta JIRA ali Asana.

## 4. Seznam zahtev in ocenjevalni list

Tabela 1 prikazuje seznam projektnih zahtev, ki vključujejo obvezne zahteve (poleg oddaje projekta) in izbirne zahteve. Za uspešno zaključene vaje je potrebno opraviti/vključiti vse obvezne zahteve in zbrati skupno najmanj 50 % točk (50 točk). Zahteve se preverjajo med zagovorom projekta in se ocenjujejo glede na obseg in kakovost implementacije. Minimalne zahteve so določene za vsakega člana skupine in se smiselno prilagodijo glede na število članov v skupini, kjer je to mogoče (npr. če je minimalna zahteva razvoj dveh mikrorstitev, je za tričlansko skupino minimalna zahteva šest mikrorstitev).

Tabela 1: Seznam zahtev.

| Postavka                                         | Opis in minimalne zahteve                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Obvezn<br>o | Točke |
|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------|
| <b>Repozitorij</b>                               | Ustvarite repozitorij za projekt z uporabo Git.<br><br>Minimalne zahteve: <ul style="list-style-type: none"><li>Nastavite Git repozitorij na platformah, kot so GitHub, GitLab ali Bitbucket.</li><li>Vključite datoteko README z informacijami o projektu (kratek opis, navodila za namestitve itd.).</li><li>Uvedite strategijo razvejitve (npr. main, dev).</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | ✓           | 1     |
| <b>Mikrorstitev in »cloud-native« aplikacija</b> | Nastavite razvojno okolje in definirajte strukturo projekta, zasnujte in implementirajte aplikacijo z uporabo arhitekture mikrorstitev, jih namestite na Kubernetes gručo ter nastavite bazo podatkov za trajno shranjevanje podatkov.<br><br>Minimalne zahteve: <ul style="list-style-type: none"><li>Uporabite IDE (npr. Visual Studio Code, IntelliJ IDEA).</li><li>Nastavite osnovno strukturo projekta v skladu z najboljšimi praksami (npr. direktoriji za izvirno kodo, teste in konfiguracijske datoteke).</li><li>Inicializirajte razvojno okolje z vsemi potrebnimi datotekami.</li><li>Dokumentirajte navodila za nastavitve za druge razvijalce.</li><li>Razdelite aplikacijo na vsaj dve mikrorstitev, kjer je vsaka odgovorna za določeno funkcionalnost.</li><li>Zagotovite, da mikrorstitev komunicirajo prek HTTP ali sporočilnega protokola.</li><li>Napišite Kubernetes manifeste (YAML datoteke) za namestitve mikrorstitev.</li><li>Uporabite funkcionalnosti Kubernetes, kot so storitve, namestitve in podi za orkestracijo.</li><li>Upoštevajte »cloud-native« koncepte.</li><li>Uporabite relacijsko (npr. PostgreSQL, MySQL) ali NoSQL (npr. MongoDB) bazo podatkov.</li><li>Zagotovite, da je baza podatkov dostopna iz mikrorstitev in mikrorstitev vanjo shranjujejo ali iz nje berejo podatke.</li><li>V podatkovno bazo dodajte nekaj tabel in podatkov.</li></ul> | ✓           | 6     |
| <b>Dokumentacija</b>                             | Ustvarite tehnično dokumentacijo za projekt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | ✓           | 2     |

|                                           |                                                                                                                                                                                                                                                                                                                                                                                                          |   |   |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---|
|                                           | <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Vključite vse relevantne informacije o aplikaciji glede na primere dobrih praks.</li> <li>Uporabite Markdown ali druga orodja za dokumentacijo za jasno organizacijo vsebine.</li> </ul>                                                                                                                                                |   |   |
| <b>Dokumentacija API</b>                  | <p>Ustvarite dokumentacijo za API.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Uporabite orodja, kot sta Swagger/OpenAPI ali Postman, za dokumentiranje končnih točk API.</li> <li>Vključite podrobnosti o formatih zahtev/odgovorov, primerih tovorov in sporočilih o napakah za vsako mikrorstitev.</li> </ul>                                                                |   | 3 |
| <b>Cevovod CI/CD</b>                      | <p>Uvedite CI/CD (nenehno integracijo in nenehno dobavo).</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Uporabite CI/CD orodje (npr. GitHub Actions, GitLab CI, Jenkins) za avtomatizacijo gradnje in testiranja.</li> <li>Zagotovite, da se koda samodejno testira in namesti na Kubernetes gručo ob spremembah v glavni veji.</li> </ul>                                        |   | 5 |
| <b>Helm charts</b>                        | <p>Uporabi Helm Charts za upravljanje Kubernetes namestitev.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Ustvarite Helm Charts za mikrorstitev.</li> <li>Parametrizirajte konfiguracije za različna okolja (razvoj, produkcija).</li> <li>Namestite in posodobite aplikacije v Kubernetes okolju s Helm Charts.</li> </ul>                                                      |   | 4 |
| <b>Namestitev v oblak</b>                 | <p>Namestite aplikacijo v oblaknem okolju.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Uporabite oblačne platforme, kot so AWS, Azure ali Google Cloud.</li> <li>Zagotovite, da je Kubernetes gruča in aplikacija javno dostopna.</li> </ul>                                                                                                                                    | ✓ | 5 |
| <b>»Serverless« funkcija</b>              | <p>Razvijte in namestite »serverless« funkcijo za del aplikacije.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Uporabite npr. AWS Lambda, Azure Functions ali Google Cloud Functions.</li> <li>Implementirajte funkcijo, ki se proži na podlagi dogodkov (npr. HTTP zahtev).</li> <li>Opazujte delovanje funkcije in prilagajajte zmogljivost na podlagi obremenitev.</li> </ul> |   | 5 |
| <b>Zunanji API</b>                        | <p>V aplikacijo integrirajte vsaj en zunanji API.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Uporabite zunanji API znotraj ene izmed mikrorstitev.</li> <li>Ustrezno izvedite potrditev verodostojnosti (ang. authentication).</li> </ul>                                                                                                                                      |   | 3 |
| <b>Večnajemništvo (ang. multitenancy)</b> | <p>Implementirajte večnajemniško arhitekturo.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Načrtujete in implementirajte večnajemništvo na nivoju mikrorstitev, Kubernetes in podatkovne baze.</li> <li>Prikažite delovanje z več najemniki.</li> </ul>                                                                                                                          |   | 5 |
| <b>Preverjanje zdravja</b>                | <p>Implementirajte preverjanja zdravja za mikrorstitev.</p>                                                                                                                                                                                                                                                                                                                                              |   | 4 |

|                                           |                                                                                                                                                                                                                                                                                                                                                                                                                  |  |   |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|---|
|                                           | <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>• Ustvarite končne točke za preverjanje zdravja, ki vrnejo stanje vsake mikrosorlitve.</li> <li>• Zagotovite, da lahko Kubernetes spremlja te končne točke za zdravje storitve.</li> </ul>                                                                                                                                                      |  |   |
| <b>GraphQL in gRPC</b>                    | <p>Uporabite GraphQL in gRPC za komunikacijo med storitvami.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>• Implementirajte GraphQL za poizvedovanje po podatkih v eni mikrosorlitvi.</li> <li>• Uporabite gRPC za komunikacijo med dvema storitvama.</li> <li>• Prikažite uporabo obeh tehnologij v aplikaciji.</li> </ul>                                                              |  | 4 |
| <b>Sporočilni sistemi</b>                 | <p>Implementirajte sporočilni sistem za komunikacijo med storitvami.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>• Uporabite sporočilnega posrednika (npr. RabbitMQ, Kafka, NATS Jetstream) za asinhrono komunikacijo med mikrosorlitvami.</li> <li>• Zagotovite, da vsaj dve mikrosorlitvi komunicirata prek sporočilnega sistema.</li> </ul>                                          |  | 5 |
| <b>»Event sourcing« in CQRS</b>           | <p>Implementirajte vzorce »Event Sourcing« in CQRS.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>• Implementirajte »Event Sourcing« za sledenje sprememb podatkov.</li> <li>• Uporabite CQRS za ločene modele branja in pisanja.</li> <li>• Demonstrirajte, kako se lahko ponovno predvajajo dogodki za obnovo stanja.</li> </ul>                                                        |  | 5 |
| <b>Centralizirano beleženje dnevnikov</b> | <p>Implementirajte centraliziran sistem za beleženje in analizo dnevnikov.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>• Uporabite orodja, kot so ELK Stack ali Fluentd za zbiranje dnevnikov.</li> <li>• Centralizirajte dnevnike vseh storitev.</li> <li>• Nastavite opozorila na podlagi zaznanih vzorcev v dnevnikih.</li> </ul>                                                    |  | 5 |
| <b>Zbiranje metrik</b>                    | <p>Implementirajte sistem za zbiranje metrik delovanja aplikacije.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>• Uporabite Prometheus, Grafana ali podobno orodje za zbiranje in vizualizacijo metrik.</li> <li>• Spremljajte ključne metrike aplikacije, kot so odzivnost in obremenitev.</li> <li>• Prikažite uporabo zbranih metrik za optimizacijo delovanja aplikacije.</li> </ul> |  | 5 |
| <b>Izolacija in toleranca napak</b>       | <p>Zagotovite izolacijo mikrosoritev in odpornost na napake.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>• Implementirajte mehanizme za toleranco napak, kot so Circuit Breaker in Retry.</li> <li>• Prikažite, kako sistem ohranja delovanje tudi ob napakah ene storitve.</li> <li>• Pojasnite, kako izolacija storitev zmanjšuje vpliv napak na celoten sistem.</li> </ul>           |  | 5 |
| <b>Upravljanje s konfiguracijo</b>        | <p>Implementirajte koncepte za upravljanja konfiguracije.</p> <p>Minimalne zahteve:</p>                                                                                                                                                                                                                                                                                                                          |  | 4 |

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                          |   |            |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|------------|
|                            | <ul style="list-style-type: none"> <li>• Identificirajte vso konfiguracijo vaše aplikacije.</li> <li>• Ločite implementacijo in konfiguracijo – konfiguracijo naj bo mogoče spreminjati brez ponovnega prevajanja in nameščanja mikrororitve.</li> <li>• Predvidite in uvedite eno uporabo različnih virov konfiguracije, npr. konfiguracijske datoteke, okoljske spremenljivke, konfiguracijski strežnik ...</li> </ul> |   |            |
| <b>Grafični vmesnik</b>    | Razvijte grafični uporabniški vmesnik (GUI) za aplikacijo.<br><br>Minimalne zahteve: <ul style="list-style-type: none"> <li>• Uporabite ogrodje (npr. React, Angular, Vue.js) za izdelavo GUI ali razvijte mobilno aplikacijo.</li> <li>• Implementirajte vsaj 2 podstrani.</li> <li>• Zagotovite, da sprednji del komunicira z mikrororitvami prek REST API-jev.</li> </ul>                                             | ✓ | 4          |
| <b>Vmesna predstavitev</b> |                                                                                                                                                                                                                                                                                                                                                                                                                          |   | 10         |
| <b>Oddaja projekta</b>     |                                                                                                                                                                                                                                                                                                                                                                                                                          | ✓ | 10         |
|                            |                                                                                                                                                                                                                                                                                                                                                                                                                          |   | <b>100</b> |

Poleg zahtev iz Tabele 1 lahko v svojo rešitev vključite tudi koncepte iz zahtev za dodatne točke (Tabela 2) in si tako zvišate oceno vaj. Te točke se ne štejejo v kvoto za opravljene vaje, najvišja možna končna ocena pri vajah pa ostaja 100 točk (100 %).

*Tabela 2: Seznam zahtev za dodatne točke.*

| Postavka                  | Opis in minimalne zahteve                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Obvezno | Točke |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------|
| <b>Terraform</b>          | Uporabite Terraform za avtomatizacijo namestitve infrastrukture v oblak.<br><br>Minimalne zahteve: <ul style="list-style-type: none"> <li>• Ustvarite Terraform konfiguracijske datoteke za definiranje potrebnih virov (npr. virtualni stroji, omrežja, baze podatkov).</li> <li>• Uporabite module za ponovno uporabo konfiguracij.</li> <li>• Implementirajte Terraform »plan« in »apply« za izpeljavo sprememb infrastrukture.</li> </ul>                                                                               |         | 3     |
| <b>API Gateway</b>        | Implementirajte API Gateway za centralizirano upravljanje dostopa do mikrororitv.<br><br>Minimalne zahteve: <ul style="list-style-type: none"> <li>• Uporabite API Gateway (npr. Kong, AWS API Gateway, Istio) za usmerjanje API klicev.</li> <li>• Ustrezno nastavite potrditev pristnosti (ang. authentication) in avtorizacijo (ang. authorization) preko API Gateway.</li> <li>• Implementirajte enotno točko dostopa za različne API-je.</li> <li>• Prikažite delovanje API Gatewa z večimi mikrororitvami.</li> </ul> |         | 4     |
| <b>Ingress Controller</b> | Ustvari Ingress Controller za upravljanje dostopa do storitev v Kubernetes.<br><br>Minimalne zahteve: <ul style="list-style-type: none"> <li>• Namestite Ingress Controller (npr. NGINX, Traefik) v svoje Kubernetes okolje.</li> <li>• Nastavite Ingress pravila za usmerjanje prometa do ustreznih storitev.</li> <li>• Implementirajte TLS/SSL za varno komunikacijo.</li> </ul>                                                                                                                                         |         | 4     |

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |  |           |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------|
|                          | <ul style="list-style-type: none"> <li>Preverite delovanje Ingress Controllerja z dostopom do storitev preko URL-jev.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                            |  |           |
| <b>IAM, OAuth2, OIDC</b> | <p>Uporabite upravljanje identitete in dostopa (IAM) sistem za upravljanje identitet in dostopa.</p> <p>Minimalne zahteve:</p> <ul style="list-style-type: none"> <li>Implementirajte IAM rešitev (npr. AWS IAM, Azure AD, Keycloak) za upravljanje uporabniških dostopov.</li> <li>Uporabite OAuth2 za avtorizacijo dostopa do API-jev.</li> <li>Implementirajte OpenID Connect (OIDC) za avtorizacijo in pridobivanje informacij o uporabnikih.</li> <li>Preverite delovanje s prijavo uporabnika in dodelitvijo pravic dostopa do aplikacije.</li> </ul> |  | 4         |
|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |  | <b>15</b> |

Ker je glavna tema tega predmeta razvoj spletnih aplikacij, optimiziranih za izvajanje v oblaku, si prizadevajte zasnovati svoje aplikacije tako, da kar najbolje odražajo realne zahteve modernih aplikacij. Pri zasnovi takšnih aplikacij se lahko pojavijo težave, katerih rešitve pa ne spadajo v okviru samega predmeta. Da bi vaše aplikacije bile zanimive, hkrati pa ne bi porabili preveč časa za implementacijo domensko-specifičnih težav, lahko nekatere funkcionalnosti v vaših mikrostoritvah nadomestite z t. i. "mock" implementacijo. Na primer, če bi bilo smiselno vključiti mikrostoritev za analizo zgodovinskih lokacijskih podatkov, jasno določite vmesnike spletnih storitev za to mikrostoritev in jo povežite z ostalimi mikrostoritvami v aplikaciji. Implementacijo analize pa lahko poenostavite do te mere, da bo še vedno očiten smisel in namen te mikrostoritve.

## 5. Predlogi tem

Pri izbiri teme za projekt se osredotočite na svoje interese in realne probleme, ki vas obkrožajo. Pomembno je, da izberete temo, ki vas navdušuje in vam omogoča kreativno razmišljanje ter praktično reševanje izzivov. Spodaj je nekaj predlogov tem, ki jih lahko uporabite kot izhodišče, vendar vas spodbujamo, da predlagate svoje ideje:

### 1. Platforma za personalizirano učenje

**Opis projekta:** Razvijte platformo za personalizirano učenje, kjer se uporabniki lahko registrirajo in prejemajo prilagojene priporočila za izobraževalne vsebine (video posnetki, članki, vaje). Platforma vključuje več storitev: eno za upravljanje izobraževalnih vsebin, drugo za obravnavo uporabniških profilov in preferenc ter tretjo za generiranje personaliziranih učnih poti na podlagi interakcij uporabnikov. Pri tem lahko uporabite Azure za namestitve in storitve za analitiko, ki priporočajo vsebine na podlagi uporabniških vzorcev branja ali gledanja in OpenAI API za ustvarjanje priporočil ali razlag.

## 2. Pametna aplikacija za sledenje fitnessu in priporočila

**Opis projekta:** Ustvarite pametno fitness aplikacijo, ki sledi podatkom o vadbi uporabnika, analizira te podatke in priporoča prihodnje načrte vadbe na podlagi uspešnosti, preferenc in ciljev. Aplikacija naj vključuje tudi socialne funkcije, kjer lahko uporabniki delijo dosežke s prijatelji ali sodelujejo v fitness izzivih. Potencialna integracija Azure AI za analizo podatkov o vadbi in generiranje pametnih fitness ciljev ter OpenAI API za razlage, predloge in motivacijska sporočila. V aplikacijo lahko vključite tudi dejanske pametne naprave (npr. ure).

## 3. Sistem za upravljanje energije v pametnem domu

**Opis projekta:** Ustvarite pametno aplikacijo, ki spremlja porabo energije v gospodinjstvu in ponuja priporočila za varčevanje z energijo. Sistem naj omogoča uporabnikom nadzor in spremljanje različnih pametnih naprav (luči, termostati, aparati) ter zagotavlja takojšnje nasvete za zmanjšanje porabe energije. Potencialna integracija Azure za oblačno namestitve in povezavo z IoT Hubom za upravljanje pametnih naprav in OpenAI API za razlago podatkov in ustvarjanje vpogledov, nasvetov za prihranek energije ali napovedovanje prihodnje porabe.

## 4. Sistem za upravljanje parkirišč v pametnih mestih

**Opis projekta:** Razvijte oblačni sistem za upravljanje parkirišč, ki uporabnikom pomaga najti, rezervirati in plačati parkirna mesta v pametnih mestih. Sistem naj se integrira s senzorji za parkiranje v realnem času in občinskimi bazami podatkov, da sledi razpoložljivosti parkirnih mest v različnih mestnih conah. Potencialna integracija in povezava z zunanjimi API-ji, ki zagotavljajo podatke o parkiranju, storitvah geolokacije (Google Maps API) in plačilnimi prehodi (npr. Stripe). Primer dostopnih podatkov: <https://www.lpt.si/parkirisca/informacije-za-parkiranje/prikaz-zasedenosti-parkirisc>

## 5. SaaS sistem za upravljanje zaloge za mala podjetja

**Opis projekta:** Ustvarite oblačni sistem za upravljanje zaloge, ki malim podjetjem omogoča spremljanje ravni zalog, upravljanje dobaviteljev in avtomatizacijo procesov naročanja. Sistem se lahko integrira z zunanjimi platformami za e-trgovino (npr. Shopify, WooCommerce) za sinhronizacijo razpoložljivosti izdelkov in samodejno posodabljanje zalog po prodaji.

## 6. Distribuirana trgovinska platforma

**Opis projekta:** Razvijte platformo za distribuirano trgovanje, ki omogoča uporabnikom nakup in prodajo različnih digitalnih sredstev. Sistem naj vključuje funkcionalnosti za upravljanje transakcij, preverjanje identitete uporabnikov in varno obdelavo plačil.



## 7. Sistem za obveščanje o poplavah in ekstremnih vremenskih pogojih

**Opis projekta:** Ustvarite sistem za obveščanje, ki uporabnikom posreduje informacije o poplavah in ekstremnih vremenskih pogojih. Sistem naj zbira podatke iz različnih virov in uporabnikom omogoča, da prejmejo obvestila glede na svojo lokacijo in trenutne vremenske razmere.

## 8. Sistem za upravljanje dogodkov

**Opis projekta:** Zgradite oblačno aplikacijo, ki uporabnikom omogoča načrtovanje in upravljanje dogodkov (npr. konference, poroke, srečanja). Sistem naj podpira funkcionalnosti, kot so rezervacija prostorov, upravljanje gostov, ustvarjanje agend in opomnikov. Potencialne zunanje integracije za rezervacijo prostorov, obdelavo plačil in pošiljanje povabil prek e-pošte ali SMS-a.

## 9. Aplikacija za dostavo hrane

**Opis projekta:** Razvijte oblačni sistem za dostavo hrane, kjer lahko uporabniki pregledujejo restavracije, naročajo hrano, sledijo dostavi v realnem času in puščajo ocene. Restavracije naj upravljajo svoje menije in prejemajo naročila strank prek ločenega vmesnika.

## 10. Organizator poti javnega prevoza (ali kolesarske poti)

**Opis projekta:** Razvijte aplikacijo za načrtovanje poti prek več povezav javnega prevoza, ki lahko optimizira povezave za prevoznike (predlaga boljše vozne rede), omogoča nakup vozovnic in obvešča o prepolnih avtobusih ter poznih odhodih. Sistem naj omogoča crowdsourcing informacij, kar uporabnikom omogoča, da prijavljajo težave in delijo izkušnje.

## 11. Iskanje bivališča ali stanovanj

**Opis projekta:** Ustvarite platformo, ki uporabnikom omogoča iskanje bivališč ali stanovanj na podlagi različnih parametrov, kot so lokacija, cena in velikost. Sistem naj uporabnikom omogoča tudi obveščanje o novih razpoložljivostih in spremembah na trgu nepremičnin. Dodatno lahko uporabnik išče in izbira državo ali kraj bivanja glede na želene parametre (npr. onesnaženost zraka).

# 6. Opravljanje vaj s seminarjem

Namesto s projektom lahko vaje opravite tudi z individualnim delom na seminarju. Seminar lahko kasneje uporabite kot osnovo za magistrsko delo.

Seminar mora obsegati približno 15 strani. Ker bo večina virov v angleščini, je seminar možno pisati v angleščini. Plagiatorstvo bomo preverjali, uporaba orodij generativne umetne inteligence, kot je ChatGPT, so dovoljena zgolj za izboljšanje berljivosti in jezika rokopisa. Struktura seminarja predvidoma vključuje naslednje elemente: naslov, ki jasno opredeljuje temo; povzetek, ki na kratko povzame ključne točke in ugotovitve; uvod, kjer opišete ozadje, motivacijo in cilje raziskave; metode za iskanje virov, kjer navedete kriterije za izbor literature in vire; pregled področja, kjer predstavite obstoječe rešitve, standarde ali koncepte; primerjavo rešitev, kjer na podlagi različnih kriterijev primerjate obstoječe pristope; metode, rešitve in vrednotenje, kjer opišete vašo rešitev in jo ovrednotite; zaključek in seznam literature, kjer navedete uporabljene vire.

Potek in mejniki pri pisanju seminarja so podobni kot pri projektu (prijave teme, neobvezna vmesna predstavitev, oddaja in končni zagovor). Pri končnem zagovoru se ocenjuje predstavitev seminarja in vsebino seminarja.

Spodaj je navedenih nekaj predlaganih tem za seminar, predlagate pa lahko tudi lastno temo:

### **1. Definicija in zbiranje večplastnih metrik**

Seminar bo raziskal možnosti in metode zbiranja večplastnih metrik, ki so pomembne za merjenje delovanja in vidljivost kompleksnih sistemov. Obravnavane bodo strategije za zajem metrik na različnih nivojih (aplikacijski nivo, infrastruktura, družba, podjetje) in njihovo uporabo za analizo zmogljivosti, stabilnosti in optimizacije sistema. Prav tako bodo predstavljeni primeri iz prakse ter tehnike za učinkovito obdelavo in vizualizacijo zbranih podatkov.

### **2. Rudarjenje in konstrukcija grafov komunikacije in odvisnosti**

Osrednja tema seminarja bo analiza komunikacije med različnimi storitvami in komponentami infrastrukture s pomočjo grafov. Predstavljene bodo tehnike rudarjenja in konstrukcije grafov, ki omogočajo vpogled v delovanje in optimizacijo povezav ter identifikacijo ozkih grl in anomalij v sistemih. Poseben poudarek bo namenjen uporabi tehnik strojnega učenja za analizo vzorcev.

### **3. Agregacija metrik in znanja na več nivojih: preučevanje federativnih pristopov, vzorčenja, pogostosti in preoblikovanja informacij**

Seminar bo obravnaval napredne pristope združevanja metrik in informacij, ki izvirajo iz različnih nivojev kompleksnih sistemov. Preučeni bodo federativni pristopi, tehnike vzorčenja in strategije za transformacijo informacij, kjer je cilj odgovoriti na vprašanje

koliko informacij potrebujemo na posameznem nivoju kompleksnih sistemov za učinkovito upravljanje in analizo.

#### **4. Dinamična rekonfiguracija zbiranja podatkov**

Seminar nadgradi predhodno temo »Agregacija metrik in znanja na več nivojih« z obravnavo metod dinamične rekonfiguracije zbiranja podatkov (npr. pri OpenTelemetry), kjer je cilj, da se sistem zajemanja podatkov samodejno prilagodi glede na potrebe sistema in cilje. Opcijsko se uporabi »swarm« pristope za učinkovito obvladovanje heterogenih in razpršenih sistemov.

#### **5. Razporejanje delovnih obremenitev: visoko- in nizko-nivojska dejanja ter strategije nameščanja**

V tem seminarju bodo analizirane različne strategije nameščanja strokov (ang. pod) in obravnavane visoko- in nizko-nivojske akcije za umestitev strokov v gruči, kjer so nizko-nivojske akcije lahko i) postavi strok na vozlišče, ii) prestavi strok iz vozlišča 1 na vozlišče 2 itd. V seminarju se analizira katere visokonivojske akcije, ki so sestavljene iz nizkonivojskih akcij, so v tem kontekstu smiselne. Seminar dodatno raziše možnosti optimizacije načrtov izvajanja akcij v primerih, ko moramo izvesti skupino akcij.

#### **6. Kodiranje konfiguracij, izvajanje, in izmenjava**

Seminar bo raziskal metode predstavitve konfiguracij za potrebe opisov ciljev ravni storitev in dogovorov o ravni storitev (SLA/SLO), njihovo udejanjanje ter strategije za izmenjavo in deljenje tovrstnih konfiguracij med različnimi sistemi. Predstavljeni bodo različni formati kodiranja konfiguracij in platforme za samodejno izvajanje teh nastavitev v različnih okoljih ([primer](#)). Obravnavane bodo tudi tehnike za zagotovitev učinkovitega deljenja.

#### **7. Vidljivost sistema: veliki jezikovni modeli, uporabniški vmesniki, zbiranje dnevnikov**

Seminar bo preučil različne napredne aspekte za vidljivost sistemov s poudarkom na naprednih uporabniških vmesnikih za vidljivost sistemov, zbiranje in analizo dnevniških zapisov, in uporabo velikih jezikovnih modelov (LLM) za ustvarjanje vizualizacij v nadzornih ploščah ter razlago metrik sistema. Predstavljeni bodo primeri uporabe LLM-jev za pomoč pri odpravljanju napak in izboljšanje uporabniške izkušnje, ter inovativni pristopi za zbiranje podatkov o delovanju sistemov.