

Kërkesat Teknike rreth projektit në lëndën Lab Course 1 - 2025

Menaxhimi i Kodit dhe Detyrave:

- **Git:** Përdorimi i Git.
- **Trello:** Përdorimi i Trello për menaxhimin e detyrave.

Backend:

- .NET Core Web API:
- Node.js
- Spring Boot
- PHP Laravel

Krijimi i një API të sigurt dhe efikas për menaxhimin e të dhënave. API duhet të mundësojë CRUD (Krijim, Lexim, Update, Fshirje) operacionet për të gjitha entitetet e sistemit.

Frontend:

- ReactJS
- Angular
- Vue.js

Në këtë pjesë duhet të implementoni gjithashtu Bootstrap apo Tailwind.

Databaza:

- MSSQL
- MySQL
- Oracle

Duhet të përdoret një sistem menaxhimi i bazave të të dhënave relacional si SQL Server, MySQL apo Oracle. Struktura e të dhënave duhet të jetë mirë e organizuar dhe të mbështesë integritetin e të dhënave.

Siguria:

- Implementimi i JWT ose teknikave të tjera të autentifikimit duhet të konsiderohet për të siguruar qasjen në API ose pjesët e mbrojtura të aplikacionit.

Kriteret e Pranueshmërisë:

- Sistemi duhet të funksionojë pa probleme në shfletuesit kryesorë (Chrome, Firefox, Safari, Brave, Edge).
- Duhet të ketë dokumentacion të qartë për API-në dhe për përdorimin e aplikacionit.
- Performanca e aplikacionit duhet të jetë e mirë.
- Sistemi duhet të jetë i sigurt dhe të mbrojtë të dhënat e përdoruesve nga akseset e paautorizuara.
- Çdo projekt duhet të implementojë Dashboard (obligative):
 - Në rast që keni implementuar **Bootstrap** (vetëm si shembull : <https://startbootstrap.com/theme/sb-admin-2>)
 - Në rast që keni implementuar **Tailwind** (vetëm si shembull : <https://www.tailwindawesome.com/resources/flowbite-admin-dashboard>)

Kriteret e Vlerësimit:

Git (5%)

- Përdorimi efektiv i Git për bashkëpunim.
- Struktura e qartë e commits-ve me mesazhe përshkruese.

Trello (5%)

- Përdorimi i Trello për menaxhimin e detyrave:
- Organizimi i kartave dhe përdorimi i etiketave për të treguar statusin e detyrave.
- Përdorimi i komenteve dhe bashkëpunimi në karta për të diskutuar dhe zgjidhur çështje.

Dizajni dhe Organizimi i Bazës së Të Dhënave (25%)

- Struktura e bazës së të dhënave është logjike dhe efikase:
- Schemat të jenë të qartë me tabelat e duhura dhe relacionet ndërmjet tyre.
- Përdorimi i indekseve dhe kufizimeve (constraints).

Backend (20%)

- Arkitektura dhe Kodimi.
- Kontrollerat janë të ndërtuara sipas praktikave më të mira dhe standardeve të kodimit.
- Përdorimi i Entity Framework (për .NET Core projekte) ose teknologji të ngjashme për manipulimin e të dhënave.
- Nën-kategori:
 - **Business logic (10%):** Implementimi është i qartë dhe efikas.
 - **Autentifikimi dhe Autorizimi (10%):** Mekanizmat e sigurisë për qasje në API ose në pjesët e ndjeshme të aplikacionit.

FrontEnd (25%)

- Dizajni dhe Performanca.
- Ndërfaqja e përdoruesit është intuitive, reaguese dhe e përshtatshme për përdorim në pajisje të ndryshme.
- Nën-kategori:
 - **Interaktiviteti (15%):** Implementimi i elementeve interaktive si formave, filtrave dhe navigimi (menu).
 - **Optimizimi (10%):** Teknikat e optimizimit si lazy loading, code splitting, etj., janë të përdorura për të përmirësuar kohën e ngarkimit të faqes.

Siguria (20%)

- Implementimi i praktikave të sigurisë:
 - **Ruajtja e Tokenëve (10%):** Nëse tokenët janë ruajtur në mënyrë të sigurtë, duke parandaluar sulmet e mundshme.
 - **Rinovimi i Tokenëve dhe Menaxhimi i Seancave (10%):** Implementimi i **refresh tokens** dhe menaxhimi i sesioneve.

Kjo formë do të përdoret për vlerësimin e projektit (në afatet e rregullta të provimeve), ndërsa për vlerësimin individual do të mbahet një test në çdo afat të provimeve, që përmban detyra rreth projektit.