

TDP003 Projekt: Egna datormiljön

Projektplan

Författare

Johan Törner, johto839@student.liu.se

Linus Nordin, linno988@student.liu.se

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Skapad	23-09-18

2 Inledning

Projektets syfte är att skapa en portfolio som innehåller de projekt som skapas under IP-programmet. Portfolion är en webbsida varifrån en användare kan söka på projekt som ligger i en databas. Ansvariga för projektet är Johan Törner och Linus Nordin. Projektet kommer att följa de riktlinjer som satts och kommer att innehålla de delar som specificerats.

3 Arbetssätt

3.1 Tekniker

De tekniker som kommer att användas i projektets gång är följande: Python, HTML, CSS, TailwindCSS, Flask, Jinja2, JSON, Git samt Latex.

3.1.1 Backend

På backend används Python, Flask, JSON samt Jinja2. Flask är ett lightweight-framework för webben baserat på Python. Tillsammans med Jinja2 kan vi rendera HTML direkt från servern. Dessutom använder vi JSON för att spara information om varje projekt varifrån användaren sedan kan hämta dem.

3.1.2 Frontend

För frontend används HTML och CSS, med vilket Tailwind används som CSS+framework. Där som tidigare nämnt HTML renderas från servern av Jinja2 med hjälp av Flask. CSS används för att styla HTML.

3.1.3 Övrigt

Git används för versionhantering. Latex används för dokument skrivning.

3.2 Kommunikation

Kommunikation sker huvudsakligen via Discord, om en inte är närvarande på campus förstås. När man träffas på morgonen upptas dagens planering och bestämmer man som gör vad. Ytterligare information om detta finns under *uppdelning av arbetsuppgifter*.

3.3 Upplägg

Huvudsakligen utförs arbetet på plats på campus, med undantag för tillfällen då jobb måste utföras under helgen (till följd av felaktigheter i planering) alternativt när någon är sjuk. För mer information om arbetssättet, se *gruppkontraktet*. För mer specifik information angående tidsplaneringen, se *planeringen*.

3.4 Uppdelning av arbetsuppgifter

Huvudsakligen delas arbetet upp. Eftersom arbetet sker på plats går det enkelt att gå igenom andras kod, alternativt svara på frågor. Vid behov, men huvudsakligen vid starten på dagen, planeras vad som ska göras samt vem som ska göra vad för dagen. Huvudsakligen utgår man från planeringen, men med att olika moment kan ha uppskattats fel planeras dessa om.

4 Planering

Planeringens huvudsyfte är att projektet ska hållas i rätt tidsfas och att projektmedlemmarna alltså har en överblick över när och vad som ska vara klart. För varje vecka markeras planerad tid, och i efterhand, åtgången tid. Dessutom markeras prioritet. Ju närmre en deadline desto högre blir prioriteten. Delar som inte är ett krav i projektet kommer att behandlas med lägre prioritet. I det fall en korrigering måste göras markeras denna prioritet med ett ? före. Detta för att förtydliga att momentet kanske inte kommer behöva göras.

4.1 Givna deadlines

Måndag 04-09: Gruppkontraktet

Tisdag 12-09: Tidsplanen

Fredag 15-09: LoFi-prototypen

Torsdag 21-09: Installationsmanualen v1, Projektplan utkast

Torsdag 28-09: Installationsmanualen, Projektplan (brister återgärdade)

Fredag 29-09: Datalagret

Torsdag 12-10: Publicering av Portfolio, Systemdokumentation v1

Torsdag 19-10: Systemdokumentation (brister återgärdade)

4.2 Milstolpar

Måndag: 25-09: Datalagrets API-funktioner klara.

Onsdag 27-09: Mainprogram till API klar.

Onsdag 02-10: Möjlighet att söka på hemsidan och få upp projekt.

Fredag 06-10: Adminpanel klar. (Låg prio)

4.3 Veckouppdelning

(11/9 - 17/9) V.37

Denna vecka: Lofi-prototypen

Övrigt: Studera Flask, Jinja2

Deadlines denna vecka:

Fredag: Lofi-prototyp

Prioritet	Uppgift	Planerad tid	Åtgången tid
1	Skapande av prototypen	12h	8h
1	Beskrivning av prototypen	1h	30min
2	Hämta info om Flask/Jinja2	2h	2h

2023-09-18

Projektplan

(18/9 - 24/9) V.38

Denna vecka: Utkast av projektplanen, Installationsmanualen

Övrigt: Datalagret

Eventuellt: Korrigering av LoFi-prototypen

Deadlines denna vecka:

Torsdag: Projektplan Utkast, Installationsmanual v1

Prioritet	Uppgift	Planerad tid	Åtgången tid
1	Utkast av projektplanen	6h	-
1	Installationsmanualen	4h	-
1?	Korrigering LoFi-prototyp	2h	-
2	Första 3 funktioner i Datalageret	12h	-

(25/9 - 01/10) V.39

Denna vecka: Datalagret, Presentationslagret

Övrigt: Studera TailwindCSS

Eventuellt: Korrigering av Projektplan, Installationsmanualen

Deadlines denna vecka:

Torsdag: Brister i projektplan och installationsmanual återgärdade.

Fredag: Datalagret

Prioritet	Uppgift	Planerad tid	Åtgången tid
1	Resterande funktioner i Datalagret	12h	-
2	Presentationslager: Indexsida	1h	-
2	Presentationslager: Listsida	6h	-
2	Presentationslager: Teknicsida	4h	-
2	Presentationslager: Projektsida	4h	-
1?	Korrigeringar	2h	-

(02/10 - 08/10) V.40

Denna vecka: Återgärda fel, Adminpanel

Deadlines denna vecka:

Inga givna deadlines

Prioritet	Uppgift	Planerad tid	Åtgången tid
1?	Återgärda fel	12h	-
2	Adminpanel	12h	-

(09/10 - 15/10) V.41

Denna vecka: Systemdokumentation, Publicering av portfolio

Deadlines denna vecka:

Torsdag: Publicering av portfolio, Systemdokumentation

Prioritet	Uppgift	Planerad tid	Åtgången tid
1	Systemdokumentationen	6h	-
1	Studera publiceringssystem	2h	-
1?	Återgårdade fel	4h	-

(16/10 - 22/10) V.42

Denna vecka: Brister Systemdokumentation

Deadlines denna vecka:

Torsdag: Brister i systemdokumentationen återgårdade

Prioritet	Uppgift	Planerad tid	Åtgången tid
1?	Återgårdade fel	6h	-

5 Innehåll

Enligt de specifikationer som givits kommer sidan att ha en Homepage, en List-sida, en Project-sida samt en Techniques-sida. Till detta, om tid finns, tillför vi en adminpanel varifrån portfolions ägare kan lägga till projekt utan att behöva gå in i databasen. Nedan följer beskrivningar för de olika delarnas funktionalitet. I övrigt kan se mer detaljerad information om frontend i *lofi-prototypen*.

5.1 Presentationslagret

5.1.1 Home

Home är sidans indexsida. Här kommer finnas en del om användaren samt en eller flera bilder. I övrigt kommer sidan att verka som den huvudsakliga sidan för navigation. Sidan är statisk och kommer inte att visa någon information från databasen.

5.1.2 List

List-sidan innehåller ett sökfält varifrån användaren kan söka på projekt, samt filtrera och ordna efter bland annat tekniker och datum. Till varje projekt som sedan visas kommer en bild att tillhöra, lite information, samt en länk till projektets git-sida. Sidan är dynamisk.

5.1.3 Teqchniques

Tekniksidan listar de tekniker som används i projekt på sidan. Användaren ska kunna klicka på en vald teknik och sedan ska de projekt som innehåller denna teknik listas på list-sidan. Sidan är dynamisk.

5.1.4 Project

Project-sidan visar ett valt projekt och mer utförlig information om användaren klickar på det från list-sidan. Sidan är dynamisk.

5.1.5 Adminpanel

Skulle tid finnas kommer en adminpanel att implementeras. Denna ska en given användare, med största sannolikhet portfolions användare, kunna logga in till. Härifrån kommer användaren att kunna ladda upp nya projekt och hantera gamla projekt. På detta sätt behöver portfolions användare inte skriva JSON-filer direkt.

5.2 Datalagret

Datalagrets uppgift är att hantera information om projekten. I detta fall ska namn, id, start och slutdatum, kurskod, kursnamn, kurspoäng, använda tekniker, en kort beskrivning, en lång beskrivning, en liten och stor bild, en gruppstorlek samt en länk till en projektsida.

Datan lagras i JSON och hanteras med Flask och skrivs ut i HTML med hjälp av Jinja2.

Ett krav är att data ska kunna redigeras direkt i JSON-filer utan att servern startas om. Detta är dock en självklarhet i det fall att en adminpanel implementeras.