LINKÖPINGS UNIVERSITET

TDP003 Projekt: Egna datormiljön

Projektplan

Författare

Johan Törner, johto839@student.liu.se
Linus Nordin, linno988@student.liu.se

INSTITUTIONEN FÖR DATAVETENSKAP

Höstterminen 2023
Version 1.1

2023-09-18

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Första versionen	23-09-20

2 Inledning

Projektets syfte är att skapa en portfolio som innehåller de projekt som skapas under IP-programmet. Portfolion är en webbsida vairfrån 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 informaton om varje projekt vairfrån användaren sedan kan hämta dem. Dessutom består backend av en API skriven i Python. Denna API används sedan av Flask för att läsa in, men också skriva ut, till databasen där projekten sparas.

3.1.2 Frontend

För frontend används HTML och CSS samt JavaScript. JavaScript kommer användas sparsamt och mestadels för att manipulera CSS. Med hjälp av Jinja2 som kommer tillsammans med Flask kan HTML-filer renderas till användaren från servern samt att variabler kan passas igenom. På så sätt skapas dynamiska sidor där informationen kan uppdateras utan att behöva ladda om hela sidan.

3.1.3 Övrigt

Git används för versionhantering. Latex används för dokumentskrivning. För mer information på hur Git används i projektet, se ***Rutiner***.

4 Planering

Planeringens huvudstyfte 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 priortet.

I det fall en korrigering måste göras markeras denna prirotet med ett ?.
Under denna vecka markeras det som måste arbetas med under veckan.
Under övrigt nämns det som borde göras men kan utelämnas om tid inte finns.
Under eventuellt läggs saker som inte finns i kravspecifikationen eller korrigeringar.

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)

4.1 Milstolpar

Måndag: 25-09: Datalagrets API-funktioner klara.
Onsdag 27-09: Mainprogram till API klar. (Backend med Flask)
Onsdag 02-10: Möjlighet att söka på hemsidan och få upp projekt.
Fredag 06-10: Adminpanel klar. (Låg prio)

4.2 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

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

Denna vecka: Utkast av projektplanen, Installationsmanualen
Övrigt: Datalagret
Evenutellt: Korrigering av LoFi-prototypen

Deadlines denna vecka:

Torsdag: Projektplan Utkast, Installationsmanual v1

Prioritet	Uppgift	Planerad tid	Åtgången tid
1	Utkast av projektplanen	6h	5h
1	Installastionsmanualen	4h	-
1?	Korrigering LoFi-prototyp	2h	-
2	Första 3 funktioner i Datalageret	12h	2h

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

Denna vecka: Datalagret, Presentationslagret
Övrigt: Studera TailwindCSS
Evenutellt: 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: Tekniksida	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?	Återgärda fel	4h	-

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

Denna vecka: Brister Systemdokumentation

Deadlines denna vecka:

Torsdag: Brister i systemdokumentationen återgärdade

Prioritet	Uppgift	Planerad tid	Åtgången tid
1?	Återgärda 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 beskrivingar för de olika delarnas funktionalitet. I övrigt kan en 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 datum och kurskoder. 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 Techniques

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 ägare, 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.

6 Risker och Riskhantering

I och med projektets gång kan risker uppstå. Den mest framträdande av dessa risker är sjukdom. Hantering av sjukdom är i dess enklaste form att den som drabbats arbetar hemmifrån. I denna projektgrupp har alla den möjligheten.

Övriga risker som kan uppstå är direkt kopplade till utvecklingen av programmet. Till exempel om problem skulle stötas på och planeringen inte längre går att hållas. Det enklaste sättet att motverka detta är att jobba förebyggande genom att ha bra marginal till de deadlines som är givna.

I de fall ett hinder i planeringen uppstår tillkallas projektgruppen. I de fall att sjukdom inträffat sker det, om möjigt, online. I andra fall tas det upp i början av dagen då gruppen träffas.

7 Rutiner

Gruppen hjälps åt att lösa problem, även om arbetet är uppdelat. Gruppen går igenom varandras kod dagligen och har en synpunkter uppmuntras man att lyfta dessa. Samtidigt, för att förhindra problem med versionshantering, ska medlemmarna pusha till remote efter dagens slut, förutsatt att arbete faktiskt har gjorts. Även dokument som detta pushas till remote. Det är också viktigt att medlemmerna meddelar varandra när de pushar samt vad de pushar. Extra viktigt är det för varje medlem att veta vad den andra håller på med. Man bör alltså inte basera allt det man jobbar med utifrån planeringen och det är därför gruppen pratar igenom dagen när de möts på morgonen.

7.1 Möten

Eftersom att gruppen är liten och i stort sett alltid jobbar fysiskt nära varandra beslutades att tradionella möten inte kommer att hållas. Det går istället mycket bra att diskutera saker när man parprogrammerar eller generellt när man sitter bredvid varandra. Samtidigt har inte gruppen behövt skapa en kravspecifikation eftersom denna redan är given.

7.2 Kommunikation

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

7.3 Plats

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***.

7.4 Uppdelning av arbetsuppgifter

Huvudsakligen delas arbetet upp. Eftersom arbetet sker på plats går det enkelt att gå igenom andras kod, alternativ 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 planerningen, men med att olika moment kan ha uppskattats fel planeras dessa om.