

TDP003 Projekt: Egna datormiljön

Projektplan

Författare

Simon Gradin, `simgr011@student.liu.se`

Robin Jensen, `robje408@student.liu.se`

1 Introduktion

Projektet ska ge användaren en produkt som lagrar samt visar upp en portfolio av färdigställda projekt. Produkten innehåller ett datalager, representationslager samt ett API(Application Programming Interface) som kommunicerar mellan dessa. Projektmedlemmarna innefattar Simon Gradin och Robin Jensen. Slut användarna är den målgrupp som vill använda sig av produkten för att visa upp tidigare projekt, detta inkluderar projektmedlemmarna. Alla projektmedlemmar är huvudansvariga. Projektet ingår i kursen TDP003 Innovativ Programmering årskurs 1.

2 Rutiner

Planering av möten sker muntligt mellan projektmedlemmarna med referens till planeringsdokumentet(se rubrik 4). Genomförandet sker på Linköpings Universitet vid muntligt bestämd tid. Alla dokument korrekturläses vid slutet av varje dag om de har bearbetats. Git används för versionshantering och säkerhetskopiering av artefakter. Programmering sker på utsatt tid enligt projektplaneringen. Alla projektmedlemmar programmerar samtidigt och arbetar med metoden parprogrammering. Kodgranskning sker under arbetsgång samt efter färdigställda delmoment. Alla projektmedlemmar ska granska koden, detta sker framför allt vid parprogrammering.

3 Milstolpar

Prioritering 1:

- Fredag V39 Funktionalitet för indata av JSON-fil till API
- Måndag V40 Funktionalitet för sökning av projekt
- Tisdag V40 Funktionalitet för utdata från API till hemsida
- Fredag V40 Konstruera hemsidans header, nav och footer
- Fredag V40 Konstruera main och css på index sidan
- Måndag V41 Konstruera html-main, css m.m. på sidor "projects/id"
- Tisdag V41 Konstruera html-main, css m.m. på sidan "list"
- Onsdag V41 Konstruera html-main, css m.m. på sidan "techniques"
- Torsdag V41 Konstruera html-main, css m.m. på sidan "error"

Prioritering 2:

- Integration av sociala medier
- Expansive sökfunktionalitet
- SEO anpassa hemsidan

4 Projektplanering

4.1 Vecka 37

Deadlines:

Torsdag: Deadline för Inlämning av planeringsdokument Text för 2

Aktiviteter:

Måndag: Arbeta på planeringsdokument – 2 timmar - Prioritering 1
Tisdag: Påbörja LoFi prototyp – 2 timmar - Prioritering 1
Onsdag: Läsa igenom och utföra Installationsmanual – 4 timmar - Prioritering 1
Onsdag: Komplettering av planeringsdokument - 4 timmar - Prioritering 1

Totalt: 12 timmar

4.2 Vecka 38

Deadlines:

Torsdag: Deadline för Inlämning av LoFi-prototyp och installationsmanualen

Aktiviteter:

Måndag: Komplettering av LoFi prototyp – 1 timmar - Prioritering 1
Tisdag: Fortsatt utveckling på installationsmanualen - 4 timmar - Prioritering 1
Onsdag: Påbörja projektplan – 3 timmar - Prioritering 1

Totalt: 8 timmar

4.3 Vecka 39

Deadlines:

Torsdag: Deadline för Inlämning av projektplan och gemensam installationsmanual

Aktiviteter:

Måndag: Arbete med projektplan – 5 timmar - Prioritering 1
Tisdag: Arbete med projektplan - 5 timmar - Prioritering 1
Onsdag: Samarbete för den gemensamma installationsmanualen - 3 timmar - Prioritering 1
Fredag: Funktionalitet för indata av JSON-fil till API - 5 timmar - Prioritering 1
Fredag: Förberedelser för kommande vecka. Läs på om tekniker bland annat. - Prioritering 2

Totalt: 18 timmar

4.4 Vecka 40

Deadlines:

Torsdag: Datalager färdigställt och inlämnat

Torsdag: Icke-trivial förbättring av gemensamma installationsmanualen.

Aktiviteter:

Måndag: Funktionalitet för sökning av projekt - 8 timmar - Prioritering 1
Tisdag: Funktionalitet för utdata från API till hemsidan - 5 timmar - Prioritering 1
Tisdag: Bidra till gemensam installationsmanual – 1 timme - Prioritering 1
Onsdag: komplettering av icke-färdigställda moment - Prioritering 2
Fredag: Konstruera hemsidans header, nav och footer - 6 timmar - Prioritering 1
Fredag: Konstruera main och css på index sidan - 2 timmar - Prioritering 1
Fredag: Förberedelser för kommande vecka. Läs på om tekniker bland annat. - Prioritering 2

Totalt: 22 timmar

4.5 Vecka 41

Deadlines:

Inga deadlines.

Aktiviteter:

Måndag: Konstruera html-main, css m.m. på sidor "projects/id" - 8 timmar - Prioritering 1

Tisdag: Konstruera html-main, css m.m. på sidor "list" - 8 timmar- Prioritering 1

Onsdag: Konstruera html-main, css m.m. på sidor "techniques" - 8 timmar - Prioritering 1

Torsdag: Konstruera html-main, css m.m. på sidor "error" - 3 timmar - Prioritering 1

Fredag: Färdigställa hemsida, fixa buggar och logiska fel, kvalitets kontroll - 7 timmar - Prioritering 1

Fredag: Uppdelning av Systemdokumentationens innehåll - 1 timme- Prioritering 2

Totalt: 35 timmar

4.6 Vecka 42

Deadlines:

Torsdag: Systemdemonstration

Torsdag: Inlämning av systemdokumentation

Torsdag: Portfolio publicerad

Aktiviteter:

Måndag: Systemdokumentation - 8 timmar - Prioritering 1

Tisdag: Komplettering av systemdokumentation och hemsidan. - Prioritering 1

Fredag: Testdokumentation - Prioritering 1

Totalt: 8 timmar

4.7 Vecka 43

Deadlines:

Torsdag: Testdokumentation inlämning, reflektionsdokument inlämning, komplettering systemdokumentation

Aktiviteter:

Måndag – Onsdag: Testdokumentation – 8 timmar - Prioritering 1

Tisdag : Komplettering av systemdokumentation – 2 timmar - Prioritering 1

Totalt: 10 timmar

5 Installationsplanen

Installationsplan för de verktyg som används i projektet "portfolio projekt" i TDP003. Innan installationerna vänligen skriv:

sudo apt update

i terminalen för att se till att all programvara är uppdaterad. Terminalen öppnas genom att klicka på Ctrl, Alt och T samtidigt.

5.1 Förutsättningar

Förutsättningar i denna manual är att användaren har ubuntu 20.04 eller senare.

5.2 Python

Python är ett programmeringsspråk som används för att programmera API:t i detta projekt. Om python inte är förinstallerat på Linux operativsystem måste detta installeras. Öppna terminalen och skriv:

sudo apt-get install python3.8 python3-pip.

För att kolla om python är installerat skrivs:

python3 --version

Om "python 3.8" visas i terminalen är installationen lyckad.

5.3 Emacs

Emacs är en texteditor som används för att underhålla samt skriva kod och andra dokument i projektet. För att installera emacs skriv kommandot:

sudo apt install emacs

Följ kommando instruktioner i terminalen.

Testa att öppna emacs genom att skriva följande i terminalen:

emacs

Om emacs startar är installationen lyckad.

5.4 Git

Git används för versionshantering samt säkerhetskopiering av projektet. Git finns som standard förinstallerat på Linux distros. För att kolla om git fungerar korrekt vänligen skriv följande i konsolen:

git

Om en manual för kommandon visas är allt som det ska. Om git inte är förinstallerat skriv:

sudo apt install git-all

Sedan testa om git är installerat igen.

5.5 GIMP

GIMP är en image-editor som används för bildredigering i projektet.

För att installera GIMP skriv följande kommando i terminalen:

sudo apt install gimp gimp-gmic

Vänligen följ följande instruktioner. För att kolla om installationen är lyckad skriv följande i terminalen:

gimp

5.6 Flask och Jinja2

Installationen av flask och jinja görs genom följande kommandon:

sudo apt-get install python-setuptools

Därefter skrev vi:

```
sudo apt install python3-pip
```

följt av

```
sudo pip3 install flask
```

För att testa om flask är korrekt installerad skriv följande i terminalen:

```
python3
```

Sedan i python3 prompten skriv:

```
import flask samt flask.__version__
```

Om flask är korrekt installerat ska versionsnummer visas i konsolen. För att testa om Jinja2 är korrekt installerat kör följande i konsolen:

```
pip show Jinja2
```

Om det är korrekt installerat ska info om versionen visas.

5.7 Firefox

Projektet utgår ifrån användning av firefox webbläsaren, vilken används för att inspektera hemsidan samt hitta information. Firefox har även inbyggda utvecklingsverktyg som används vid felsökning m.m.. Om inte Firefox är förinstallerat på operativsystemet vänligen använd följande kommando i terminalen:

```
sudo apt install firefox
```

För att vara säker på att firefox är installerat skriv följande i terminalen:

```
firefox
```

6 Hantering av artefakter

Versionshantering för alla artefakter görs via Git. Artefakter innefattar kod, dokument eller annan typ av digitalt material som ingår i projektet. Efter varje dag som projektet har utvecklats laddas det upp till gitlabs servrar. Vid start av arbetstillfälle laddas projektet ned för att vara uppdaterad till den senaste versionen.

7 Risker

Vid insjuknande av en eller flera projektmedlemmar skall arbetet fortsätta i bästa mån. Beroende på sjukdom kan de mesta arbete utföras hemifrån via röstchatt. För att förebygga mot förseningar har en projektplanering etablerats där givna moment skall vara klara i god tid före bestämda deadlines. Om projektmedlemmarna uppskattar att projektet inte utvecklas i takt med projektplaneringen kommer kursledare för TDP003 kontaktas.

8 Plan för kvalitetssäkring

Under projektets gång kommer verktyg användas för att försäkra projektets kvalitet. W3C validation Service kommer användas för att validera html kod. Givna testfiler från kursledare på kursen TDP003 kommer användas för att validera API:ns funktionalitet.