

TDP003 Projekt: Egna datormiljön

Testdokumentation

Författare

Daniel Huber, danhu849@liu.se
Jens Öhrnell, jenoh242@liu.se

Innehåll

1	Revisionshistorik	1
2	Information om denna mall	1
3	Valideringsprogram	1
4	Testspecifikation	2
4.1	Tester mot presentationslagrets krav	2
4.2	Tester mot datalagrets krav	4
4.3	Tester mot Icke-funktionella Krav	5
5	Testlogg	7

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.1	Testdokumentation Portfolio TDP003	211020
1.0	Mall för Testdokumentation Portfolio TDP003	181020

2 Information om denna mall

Författare av dokument som baseras på denna mall är införstådda med reglerna för dess användande. Reglerna återfinns i detta stycke. Varje dokument som är en påbyggnation eller använder delar av detta dokument eller någon av dess senare eller tidigare versioner ska inkludera detta stycke.

Individuella påbyggnationer eller omskrivningar av denna mall förutsätts ha indata och resultat specificerade specifikt för det egna portfolioprojektet. Endast upphovsrättsmannen, Daniel Huber (danhu849) och personer listade nedanför får använda denna mall. Dokumentet får ej delas till andra eller tredje part. Förbrytelser skickas till Diciplininnämnden vid Linköpings Universitet.

- Jens Öhrnell, jenoh242
- Michael Lake, micla389
- Robin Edlund, robed441
- Jim Teräväinen, jimte145
- Ahmed Sikh, ahmsi881

Detta samarbete har gjorts möjlig efter mejlkonversation med Examinator för Kursen TDP003, Filip Strömbäck Fredagen 16:e Oktober 2020. Frågor rörande överenskommelsens validitet hänvisas till Filip Strömbäck.

3 Valideringsprogram

För att uppfylla kraven om korrekt JSON, UTF-8 och Jinja2 i projektet har validerare skrivits i python. JSON valideras av JSON_tester.py, UTF-8 valideras av utf-8_tester.py och Jinja2 valideras av jinja2_validator.py. Även ett program som kontrollerar att alla programfiler skrivits på engelska har skrivits. Koden för samtliga program återfinns i projektrepot.

4 Testspecifikation

Nedan följer tester i ordningen x.n där x motsvaras av det övergripande kravstycket i Universitets **Systemspecifikation** och n representerar testet i ordningen. Kravet testet uppfyller visas därefter.

Indatan presenteras alltid först. Undertill visas det verkliga resultatet av senaste portfolioversionen ibland av en beskrivning, ibland av en bild och ibland av både och.

4.1 Tester mot presentationslagrets krav

Test 1.1 (Krav 1.1)

Indata Öppna portfolion första sida (URL: /).

Resultat

Test 1.2 (Krav 1.2)

Indata Öppna portfolio projektlista (URL: /list).

Resultat

Test 1.3 (Krav 1.2)

Indata Sökning i sökfältet. Endast text, exakt inmatning: '[valfritt projektnamn]' i sidan '/list'.

Resultat

Test 1.4 (Krav 1.2)

Indata Sökning i sökfältet. Endast text, exakt inmatning: 'python' på sidan '/list'.

Resultat

Test 1.5 (Krav 1.2)

Indata Klicka på 'ascending' och sedan 'Search' på sidan '/list'.

Resultat

Test 1.6 (Krav 1.2)

Indata Ändrar sorteringen från 'fallande' till 'stigande' feter sökning på '/list' med fler än ett resultat. Trycker sedan enter.

Resultat

Test 1.7 (Krav 1.2)

Indata På /list. Ändrar search_field I URL:en till något som inte finns. Trycker enter.

Resultat

Test 1. (Krav 1.2)

Indata Markerar alla searchfields genom att klicka på dem. Skriver sedan 'e' i sökfältet och trycker sedan enter.

Resultat

Test 1.8 (Krav 1.2)

Indata Klicka i 'checkbox' för 'Python' samt 'ada' och sedan 'Search' på sidan '/list'.

Resultat

Test 1.9 (Krav 1.3)

Indata Klicka på ett projekt i listan på sidan '/list'

Resultat

Test 1.10 (Krav 1.3)

Indata Ändrar URL på projektsidan från 'project/id=3' till 'project/id=4'.

Resultat

Test 1.11 (Krav 1.3)

Indata Ändrar URL på projektsidan från 'project/3' till 'project/4'.

Resultat

Test 1.12 (Krav 1.4)

Indata Markerar alla tekniker på '/techniques' sidan. Trycker sedan enter.

Resultat

Test 1.13 (Krav 1.5)

Indata Kontrollera att sökresultaten på sidan '/list/?search+projects=python&sort_by=start_date&sort_order=asc' har små bilder med w3 validator 'https://validator.w3.org'.

Resultat

Test 1.14 (Krav 1.5)

Indata Kontrollera att projektsidan '/project/1/' har minst en stor bild genom att inspektera element på sidan.

Resultat Inspektera element ger:

Test 1.15 (Krav 1.6)

Indata Efter valfri sökning på '/list'. Sätt 'search_field' variabeln i URL:en till 'IKAsm32105,saölf'.

Resultat

Test 1.16 (Krav 1.7)

Indata Ändra URL:en på projektsidan 'project/3' till 'project/a'.

Resultat

Test 1.17 (Krav 1.7)

Indata Ändra URL:en på projektsidan 'project/3' till 'project/854965625'.

Resultat

4.2 Tester mot datalagrets krav**Test 2.1 (Krav 2.1)**

Indata Testas med data__testet mot kraven, det vill säga load_test och test_get_project.

Resultat

Test 2.2 (Krav 2.2)

Indata Testas med data__testet mot kraven, det vill säga test_get_project_count och test_get_project.

Resultat

Test 2.3 (Krav 2.3)

Indata Testas med data__testet mot kraven, det vill säga test_get_techniques

Resultat

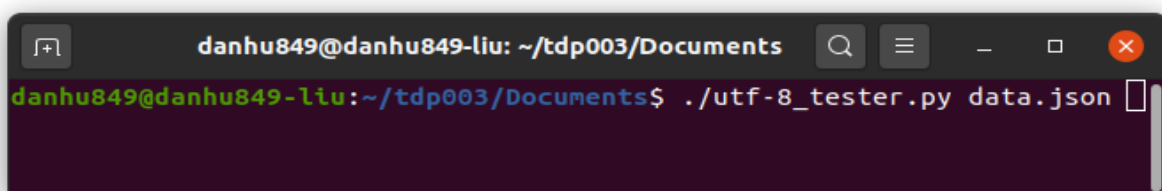
Test 2.4 (Krav 2.4)

Indata Testas med data__testet mot kraven, det vill säga test_search.

Resultat

Test 2.5 (Krav 2.7)

Indata:



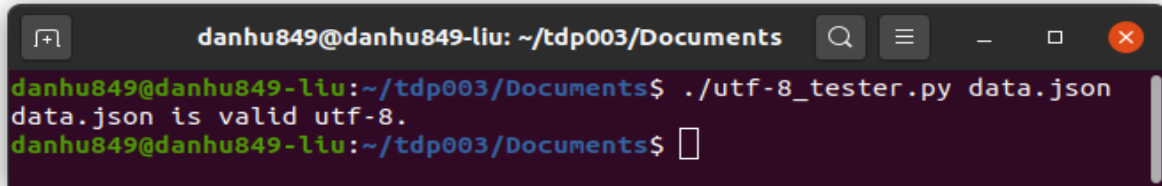
Figur 1: Hur programmet utf-8_tester.py körs i terminalen.

Resultat:

Test 2.6 (Krav 2.9)

Indata flask session startas utan debug_mode och ett femte projekt läggs till manuellt i data.json

Resultat

A terminal window with a dark background. The title bar shows the user 'danhu849@danhu849-liu' and the directory '~/tdp003/Documents'. The terminal text shows a command prompt followed by the command './utf-8_tester.py data.json', which returns the output 'data.json is valid utf-8.' followed by another prompt.

```
danhu849@danhu849-liu: ~/tdp003/Documents
danhu849@danhu849-liu:~/tdp003/Documents$ ./utf-8_tester.py data.json
data.json is valid utf-8.
danhu849@danhu849-liu:~/tdp003/Documents$
```

Figur 2: Resultat från att köra utf-8_tester.py

Test 2.7 (Krav 2.8)

Indata Lägg till ett projekt eller ett till fält på ett projekt i din JSON fil, exempel - 'Företagsnamn'. Testa sedan filen genom att köra programmet, d.v.s hemsidan. Sök på projekten med nya de fälten/fält som du har lagt till.

Resultat

Test 2.8 (Krav 2.)

Indata

Resultat

4.3 Tester mot Icke-funktionella Krav**Test 3.1 (Krav 3.1)**

Indata Validera Jinja2 i samtliga HTML filer i '/templates' genom att köra dem genom en Jinja2 parser.

Resultat

Test 3.2 (Krav 3.1)

Indata Validera att pythonscriptens utdata formateras rätt av Jinja2 i samtliga HTML filer i '/templates' med pythonscript.

Resultat

Test 3.3 (Krav 3.2)

Indata Validera portfoliosidans css3 med hjälp av w3 css validerare (<https://jigsaw.w3.org/css-validator/>). Sätt 'Profile: CSS level 3', 'Medium: All', 'Warnings: Normal report', 'Vendor Extensions: Default'.

Resultat

Test 3.4 (Krav 3.2)

Indata Validera portfoliosidans HTML5 med hjälp av w3 HTML5 validerare (<https://jigsaw.w3.org/css-validator/>).

Resultat

Test 3.5 (Krav 3.3)

Indata I terminal: cd till projektets katalog. Skriv ut katalogens innehåll med 'tree .'

Resultat

Test 3.6 (Krav 3.4)

Indata Bevisa att projektet versionhanteras med git genom att visa print screen över commit historik från portfolions repo.

Resultat

Test 3.7 (Krav 3.5)

Indata Bevisa att presentationen av systemet är godkänd och lägg till kommentarer från användare från klassens portfolio presentation.

Resultat

Test 3.8 (Krav 3.6)

Indata Kör varje python docstring från varje .py fil genom check_if_english.py

Resultat

Test 3.9 (Krav 3.7)

Indata Kör alla pythonprograms rader genom ett check_if_english.py som räknar orden och kollar om de är engelska.

Resultat

Test 3.10 (Krav 3.8, 3.9 och 3.10)

Se dokumentet: `Systemdokumentationen`

5 Testlogg

Datum	Commit	Godkända	Avvikande	Kommentar
2019-10-15			-	-
2019-10-15				12 projekt kommer upp istället för de förväntade 4. Måste åtgärdas snarast
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				
2019-10-15				