

TDP003 Projekt: Egna datormiljön

Testdokumentation

Författare

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

Innehåll

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.1	Testdokumentation Portfolio TDP003	221020
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 (danh849) och personer listade nedanför får använda denna mall. Dokumentet får ej delas till andra eller tredje part. Förbrytelser skickas till Diciplinennämnden vid Linköpings Universitet.

- Jens Öhrnell, jenh242
- 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 Testspekifikation

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 för presentationslagrets krav

Test 1.1 (Krav 1.1)

Indata Öppna portfolion första sida (URL: /).

Resultat Möts av en header med länkar till: Home, Tekniker och Avancerad Sökning. Nedanför finns sidtitel, en stor bild, introduktionstext, sociala media länkar samt footer med författar info och länk till upphovsrättshavaren av bakgrundsbilden.

Test 1.2 (Krav 1.2)

Indata Öppna portfolio projektlista (URL: /list).

Resultat Dirigeras till /list sidan och möts av sökfältsknappar, sökbar, en sökknapp och två sorteringslistor. Finns också footer med författar info och länk till upphovsrättshavaren av bakgrundsbilden.

Test 1.3 (Krav 1.2)

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

Resultat Project number 3: NEJ visas som sökresultat.

Test 1.4 (Krav 1.2)

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

Resultat Project number 1, 2 och 3 visas som sökresultat.

Test 1.5 (Krav 1.2)

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

Resultat 4 projekt visas med stigande datum från vänster till höger.

Test 1.6 (Krav 1.2)

Indata Ändrar sorteringen från 'fallande' till 'stigande' efter sökning på '/list' med fler än ett resultat. Sorteras på datum. Sedan trycks enter.

Resultat 4 projekt visas med stigande datum från höger till vänster.

Test 1.7 (Krav 1.2)

IndataFörst söks det efter python i sökfältet techniques_used på /list. techniques_used i URL:en ändras sedan till enstorkorv. Enter trycks.

Resultat Internal Server Error

Test 1.8 (Krav 1.2)

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

Resultat 10 projekt listas i sökresultaten.

Test 1.9 (Krav 1.2)

Indatapython samt ada markeras på /techniques och sedan klickas det på 'Sök'-knappen.

Resultat Projekt nummer 2 visas i sökresultaten.

Test 1.10 (Krav 1.3)

Indata Klicka på ett projekt 1 i listan på sidan '/list'

Resultat Projektsidan för projekt 1 visas.

Test 1.11 (Krav 1.3)

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

Resultat Omdirigeras till 404 sida med texten ERROR 404 The page you are trying to reach does not exist.

Test 1.12 (Krav 1.3)

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

Resultat Projektsidan byts om från projektsidan för projekt 3 till projektsidan för projekt 4.

Test 1.13 (Krav 1.4)

Indata Alla tekniker på '/techniques' sidan markeras. Sedan trycks enter.

Resultat Inga projekt visas.

Test 1.14 (Krav 1.5)

Indata Kontrollera att ett sökresultat på sidan /list har små bilder genom att inspektera element.

Resultat Inspektion av element ger: Bilden är 150px stor.

Test 1.15 (Krav 1.5)

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

Resultat Inspektion av element ger: Bilden är 300px stor.

Test 1.16 (Krav 1.6)

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

Resultat Internal Server Error

Test 1.17 (Krav 1.7)

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

Resultat Sidan omdirigeras till en sida med texten ERROR 404 The page your are trying to reach does not exist.

Test 1.18 (Krav 1.7)

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

Resultat Sidan omdirigeras till en sida med texten ERROR 404 Project with project id: 854965625 does not exist.

4.2 Tester för datalagrets krav

Test 2.1 (Krav 2.1)

Indata Testas med data_test.py 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

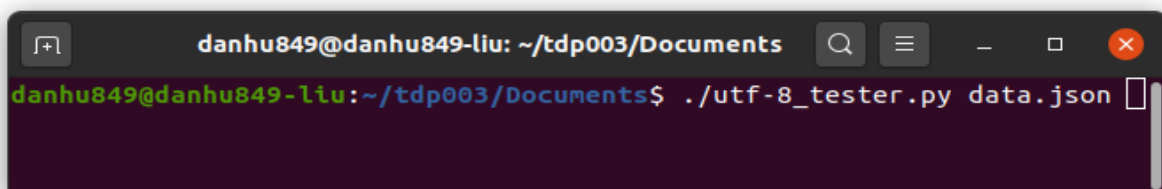
Test 2.4 (Krav 2.4)

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

Resultat

Test 2.5 (Krav 2.7)

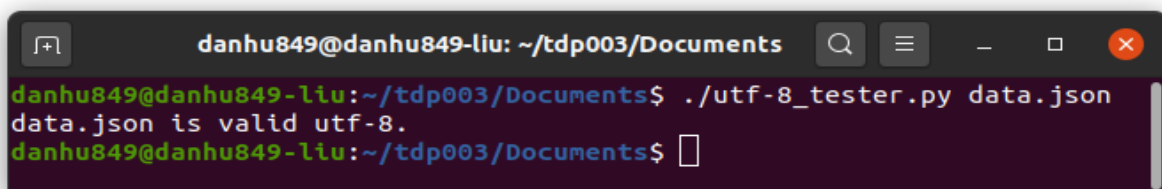
Indata:



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

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

Resultat:



```
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 2.6 (Krav 2.9)

Indata flask session startas utan debug_mode och ett femte projekt läggs till manuellt i data.json. Sedan öppnas /list i webbläsaren.

Resultat

Test 2.7 (Krav 2.8)

Indata Lägg till ett projekt eller ett till sökfält 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.)

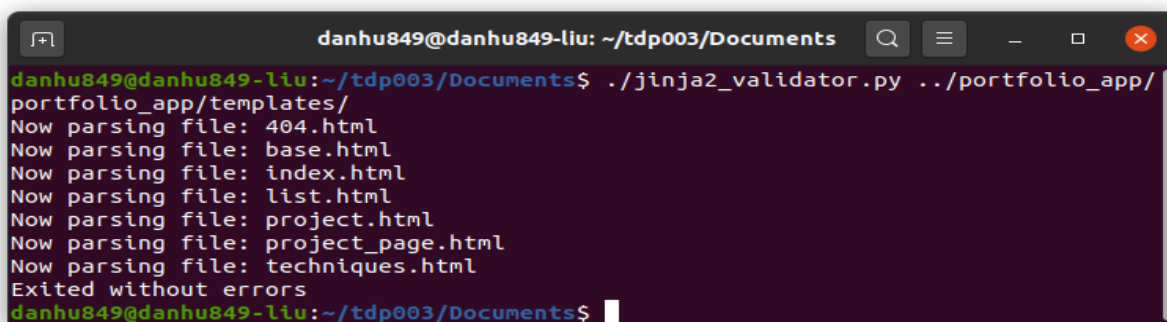
Indata

Resultat

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

Indata Validera Jinja2 i samtliga HTML filer i '/templates' genom att köra Jinja2 valideraren: jinja2_validator.py med /templates mappen som argument i terminalen:

Resultat:



```
danhu849@danhu849-liu: ~/tdp003/Documents
danhu849@danhu849-liu:~/tdp003/Documents$ ./jinja2_validator.py ../portfolio_app/
portfolio_app/templates/
Now parsing file: 404.html
Now parsing file: base.html
Now parsing file: index.html
Now parsing file: list.html
Now parsing file: project.html
Now parsing file: project_page.html
Now parsing file: techniques.html
Exited without errors
danhu849@danhu849-liu:~/tdp003/Documents$
```

Figur 3: Resultat från att köra jinja2_validator.py

Test 3.3 (Krav 3.2)

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

Resultat:

By URI By file upload By direct input

Validate by URI

Enter the URI of a document (HTML with CSS or CSS only) you would like validated:

Address:

More Options

Profile: Medium:

Warnings: Vendor Extensions:

Check

Figur 4: Resultat från att köra w3:s css3 validerare.

Test 3.4 (Krav 3.2)

Indata Validera portfoliosidans HTML5 med hjälp av w3:s HTML5 validerare (https://validator.w3.org/#validate_by_

Validate by URI Validate by File Upload Validate by Direct Input

Validate by URI

Validate a document online:

Address:

More Options

Character Encoding: Only if missing

Document Type: Only if missing

☒ List Messages Sequentially ☐ Group Error Messages by Type

☒ Show Source ☐ Clean up Markup with HTML-Tidy

☐ Show Outline ☒ Validate error pages ☐ Verbose Output

Check

Figur 5: Inställningar för w3 HTML5 validering.

Resultat:

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <http://tdp003-2020-danhu849.kubernetes-public.it.liu.se/>

Checker Input

Show ☒ source ☐ outline ☐ image report Options...

Check by

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering 4 messages hidden by filtering

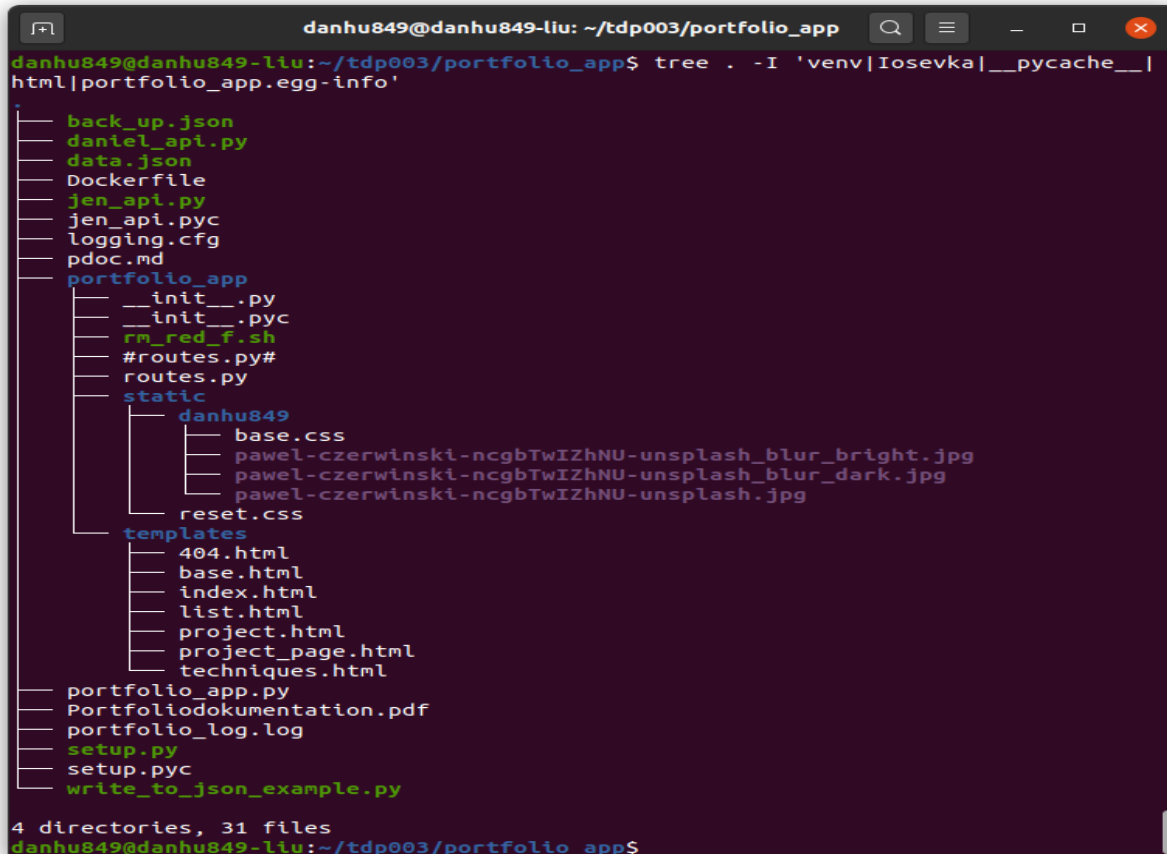
Document checking completed. No errors or warnings to show.

Figur 6: Resultat av w3 HTML5 validering.

Test 3.5 (Krav 3.3)

Indata I terminalen: cd till projektets katalog. Skriv ut katalogens innehåll med tree kommandot som syns på resultatbilden.

Resultat:



```
danhu849@danhu849-liu: ~/tdp003/portfolio_app
danhu849@danhu849-liu:~/tdp003/portfolio_app$ tree . -I 'venv|Iosevka|__pycache__|html|portfolio_app.egg-info'
.
├── back_up.json
├── daniel_api.py
├── data.json
├── Dockerfile
├── jen_api.py
├── jen_api.pyc
├── logging.cfg
├── pdoc.md
├── portfolio_app
│   ├── __init__.py
│   ├── __init__.pyc
│   ├── rm_red_f.sh
│   ├── #routes.py#
│   ├── routes.py
│   └── static
│       ├── danhu849
│       │   ├── base.css
│       │   ├── pawel-czerwinski-ncgbTwIZhNU-unsplash_blur_bright.jpg
│       │   ├── pawel-czerwinski-ncgbTwIZhNU-unsplash_blur_dark.jpg
│       │   └── pawel-czerwinski-ncgbTwIZhNU-unsplash.jpg
│       └── reset.css
├── templates
│   ├── 404.html
│   ├── base.html
│   ├── index.html
│   ├── list.html
│   ├── project.html
│   ├── project_page.html
│   └── techniques.html
├── portfolio_app.py
├── Portfoliodokumentation.pdf
├── portfolio_log.log
├── setup.py
├── setup.pyc
└── write_to_json_example.py

4 directories, 31 files
danhu849@danhu849-liu:~/tdp003/portfolio_app$
```

Figur 7: Resultat från att trädkommandot tree i terminalen.

Test 3.6 (Krav 3.4)

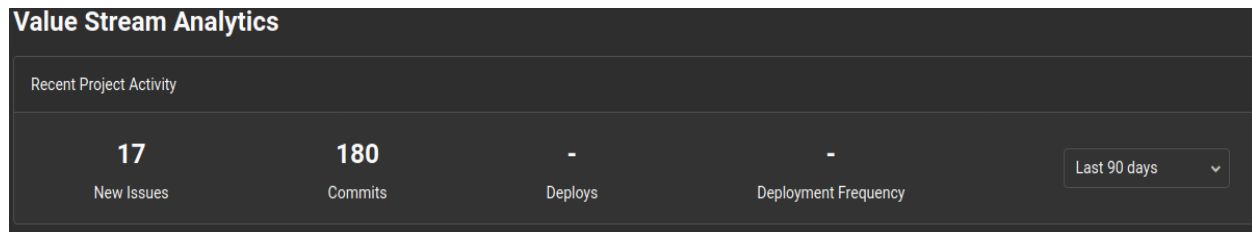
Indata Bevisa att projektet versionhanteras med git genom att visa print screen över repots 'Value Stream Analytics'.

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



Figur 8: Print Screen över Value Stream Analytics portfolio repots Analytics tab.

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 cd till /tdp003/Documents. chmod check_if_english.py om det behövs för att göra den exekverbar.
Kör programmet: ./check_if_english.py../portfolioapp/**Resultat**

Test 3.10 (Krav 3.8, 3.9 och 3.10)

Se dokumentet danhu849_jenoh242_systemdokumentation.pdf i portfolio repo <https://gitlab.liu.se/jenoh242/tdp003/>

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				