

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	2
4	Testspecifikation	2
4.1	Tester för presentationslagrets krav	2
4.2	Tester för datalagrets krav	4
4.3	Tester för Icke-funktionella Krav	6
5	Testlogg	9
A	Data.json	10
B	Feedback - Presentation av Portfolio	12

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 (danhu849) och personer listade nedanför får använda denna mall. Dokumentet får ej delas till andra eller tredje part. Förbrytelser skickas till Diciplinnämnden 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 av Daniel Huber. Frågor hänvisas till densamme. 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 program-filer skrivits på engelska har skrivits. Koddokumentation återfinns i mappen `/tdp003/Documents/validation_documentation`.

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)

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

Resultat Det står: Searching for python in search fields: enstorkorv, generated no results.

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 Alla projekt visas i sökresultaten.

Test 1.9 (Krav 1.2)

Indata python 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:

```
<img src='https://via.placeholder.com/300' alt='XXX' style='float: left;' data-module='' test='' script=''> Bilden är 300px stor.
```

Test 1.16 (Krav 1.6)

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

Resultat Följande text visas: Searching for in search fields: IKAsm32105,saölf, generated no results.

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'.

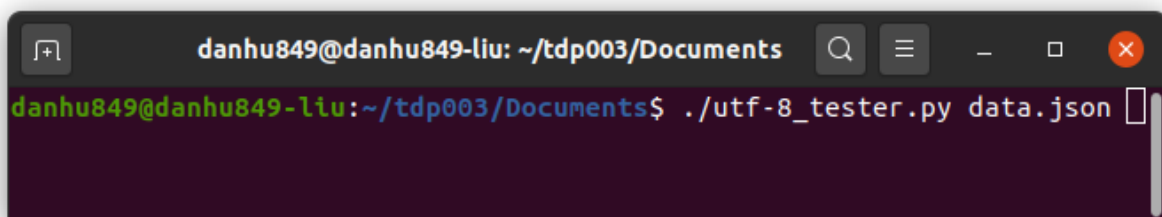
Resultat Sidan omdirigeras till en sida med texten ERROR 404 Project with project id: 854965625 does not exist.

4.2 Tester för datalagrets krav

Test 2.1 (Krav 2.1, 2.2, 2.3, 2.4)

Indata Ändra import datalager as data till import daniel_data as data i filen data_test.py. Skriv ./data_test.py i terminalen.

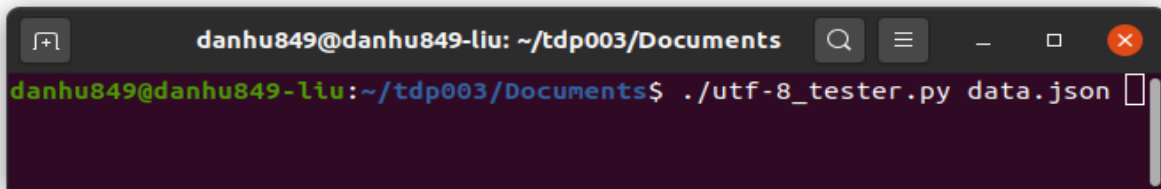
Resultat:



Figur 1: Resultatet av data_test.py körning.

Test 2.2 (Krav 2.7)

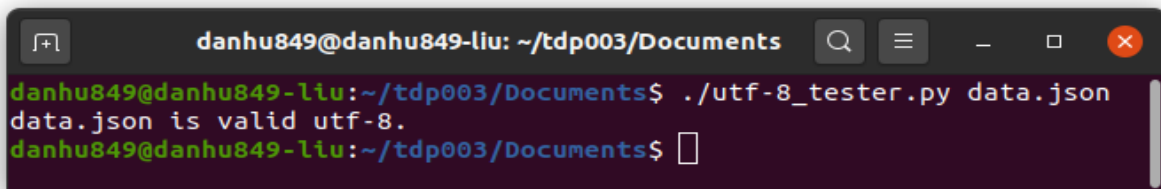
Indata:



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

Figur 2: 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 3: Resultat från att köra utf-8_tester.py

Test 2.3 (Krav 2.8)

Indata Lägg till ett projekt med ett till sökfält, 'new_search_field' i data.json. Se Appendix A. Testa sedan att köra hemsidan med flask run. Sök på 'new' med sökfältet 'new_search_list' markerat i /list.

Resultat Bara det tillagda projektet ges som sökresultat.

Test 2.4 (Krav 2.9)

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

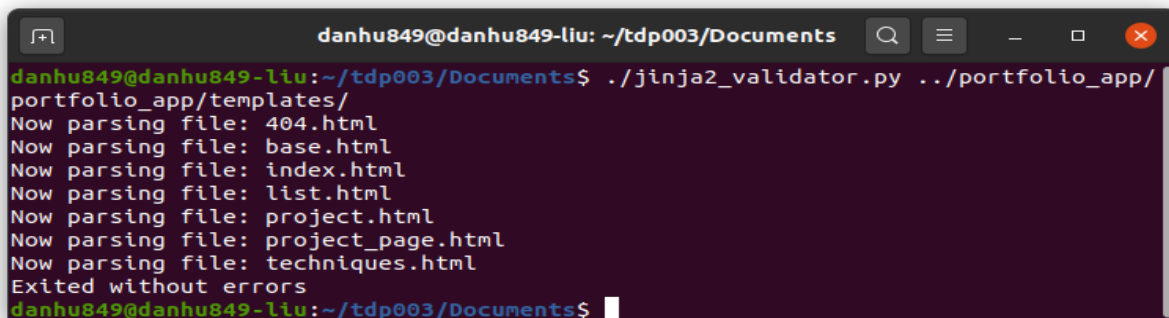
Resultat Alla fem projekt laddas in.

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 4: Resultat från att köra jinja2_validator.py

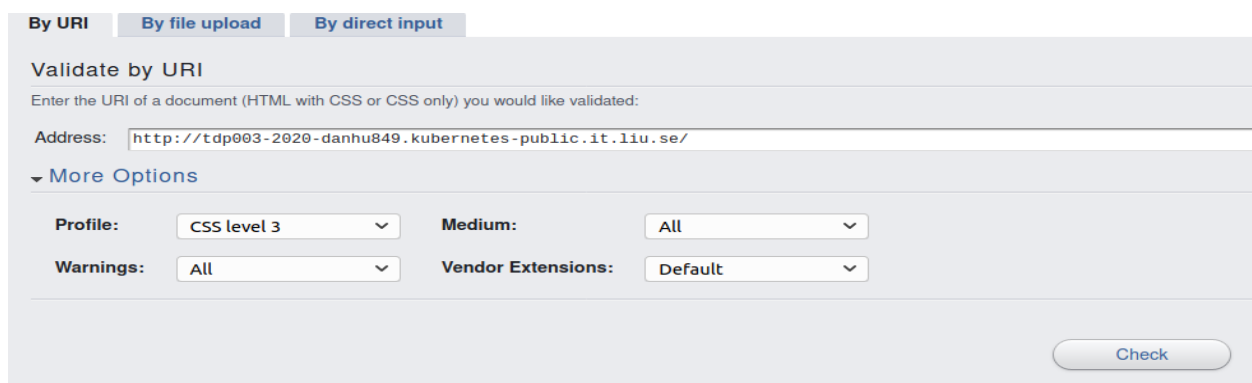
Test 3.3 (Krav 3.2)

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

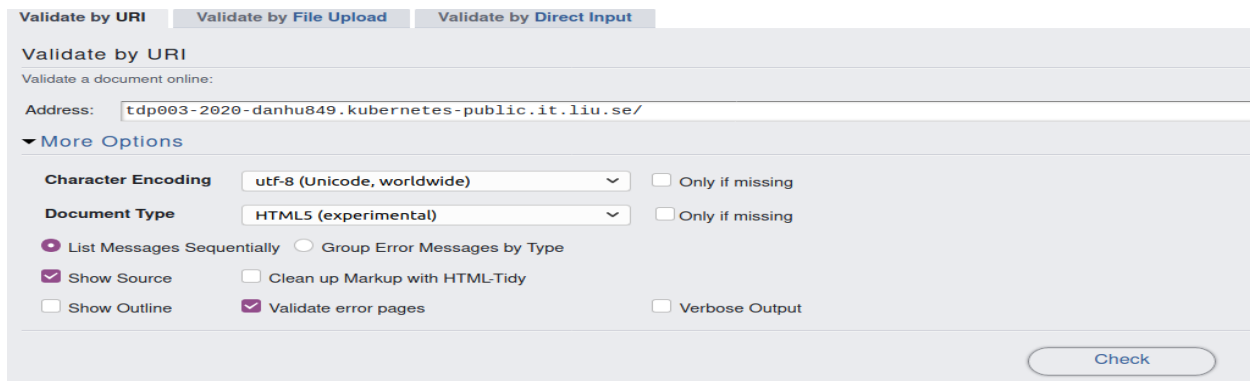
Profile: Medium:

Warnings: Vendor Extensions:

Figur 5: Resultat från att köra w3:s css3 validerare.

Test 3.4 (Krav 3.2)

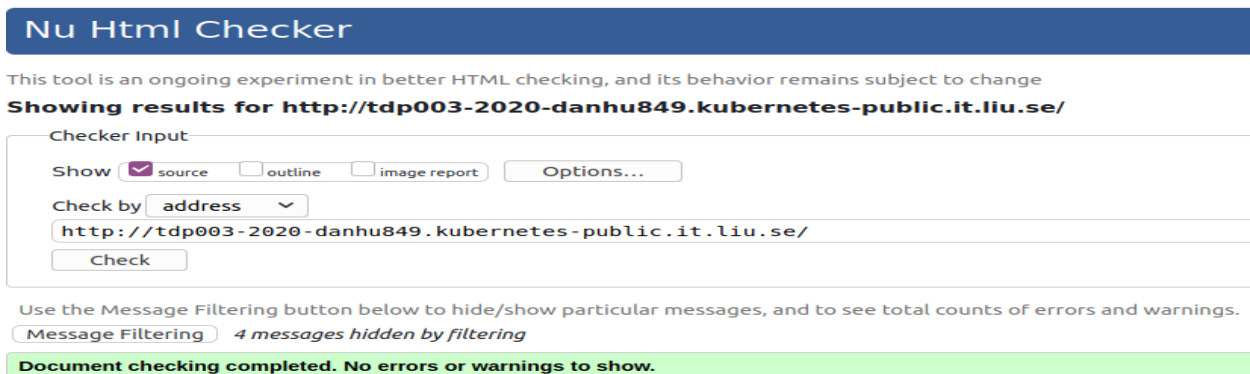
Indata Validera portfoliosidans HTML5 med hjälp av w3:s HTML5 validerare:
https://validator.w3.org/#validate_by_uri+with_options



The screenshot shows the 'Validate by URI' tab of the W3C HTML5 validator. The address field contains 'tdp003-2020-danhu849.kubernetes-public.it.liu.se/'. Under 'More Options', 'Character Encoding' is set to 'utf-8 (Unicode, worldwide)', 'Document Type' is 'HTML5 (experimental)', and 'List Messages Sequentially' is selected. Other options like 'Show Source', 'Clean up Markup with HTML-Tidy', 'Validate error pages', and 'Verbose Output' are also visible.

Figur 6: Inställningar för w3 HTML5 validering.

Resultat:



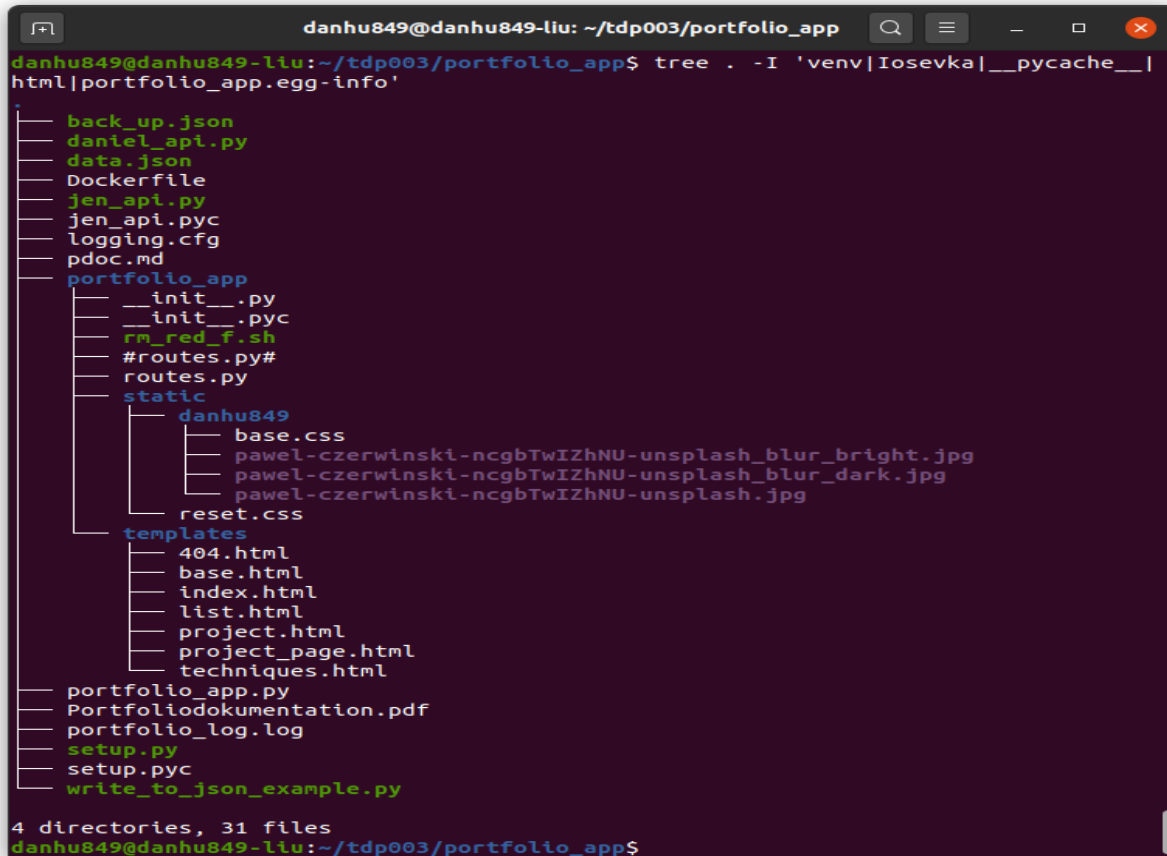
The screenshot shows the 'Nu Html Checker' results page. It displays the URL 'http://tdp003-2020-danhu849.kubernetes-public.it.liu.se/' and indicates that the document checking is completed with no errors or warnings to show. A message filtering section shows '4 messages hidden by filtering'.

Figur 7: 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 8.

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

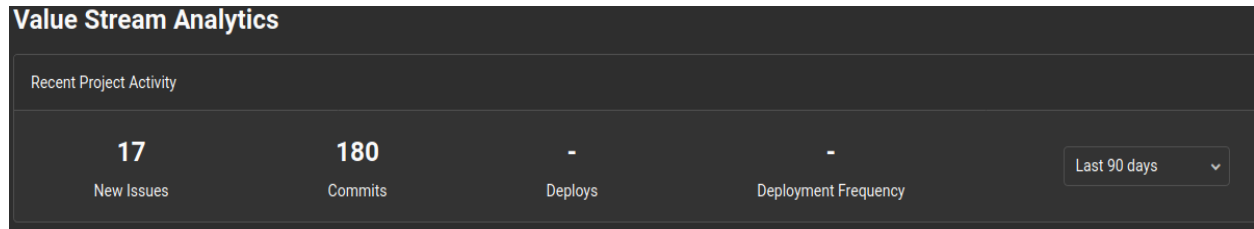
4 directories, 31 files
danhu849@danhu849-liu:~/tdp003/portfolio_app$
```

Figur 8: 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:



Figur 9: Print Screen över Value Stream Analytics portfolio repots Analytics tab.

Test 3.7 (Krav 3.5)

Indata Lägg till kommentarer från användare från klassens portfolio presentation.

Resultat Se Figur 10 i Appendix.

Test 3.8 (Krav 3.6, 3.7)

Indata cd till /tdp003/Documents. chmod check_if_english.py om det behövs för att göra check_if_english.py exekverbar. Kör programmet i terminalen: ./check_if_eng.py wordlist_eng.txt ../portfolio_app/ Skriv ut innehållet för alla filer med cat uword_directory/* och kontrollera att de okända orden endast är programmeringstermer.

Resultat

Test 3.9 (Krav 3.8, 3.9 och 3.10)

Se dokumentet danhu849_jenoh242_Systemdokumentation.pdf i portfolions repo:
<https://gitlab.liu.se/jenoh242/tdp003/-/tree/master/Documents>

5 Testlogg

Datum	Commit	Godkända	Avvikande	Kommentar
2019-10-22	bfc02812	Alla	-	-
2019-10-21	13926161	-	1.7, 1.8, 1.16	10 projekt kommer upp istället för de förväntade 4.

A Data.json

Listing 1: Utskrift av innehållet i filen data.json

```
1 [
2   {
3     "start_date": "2009-09-05",
4     "short_description": "no",
5     "course_name": "OK\u00c4NT",
6     "long_description": "no no no",
7     "group_size": 2,
8     "academic_credits": "WUT?",
9     "lulz_had": "many",
10    "external_link": "YY",
11    "small_image": "X",
12    "techniques_used": [
13      "python"
14    ],
15    "project_name": "python data-module test script",
16    "course_id": "TDP003",
17    "end_date": "2009-09-06",
18    "project_id": 1,
19    "big_image": "XXX"
20  },
21  {
22    "start_date": "2009-09-07",
23    "short_description": "no",
24    "course_name": "OK\u00c4NT",
25    "long_description": "no no no",
26    "group_size": 4,
27    "academic_credits": "WUT?",
28    "lulz_had": "few",
29    "external_link": "YY",
30    "small_image": "X",
31    "techniques_used": [
32      "c++",
33      "csv",
34      "python"
35    ],
36    "project_name": "NEJ",
37    "course_id": "TDP003",
38    "end_date": "2009-09-08",
39    "project_id": 3,
40    "big_image": "XXX"
41  },
42  {
43    "start_date": "2009-09-08",
44    "short_description": "no",
45    "course_name": "OK\u00c4NT",
46    "long_description": "no no no",
47    "group_size": 6,
48    "academic_credits": "WUT?",
```

```
49     "lulz_had": "medium",
50     "external_link": "YY",
51     "small_image": "X",
52     "techniques_used": [
53         "ada",
54         "python"
55     ],
56     "project_name": "2007",
57     "course_id": "TDP003",
58     "end_date": "2009-09-09",
59     "project_id": 2,
60     "big_image": "XXX"
61 },
62 {
63     "start_date": "2009-09-06",
64     "short_description": "no",
65     "course_name": "HOHO",
66     "long_description": "no no no",
67     "group_size": 8,
68     "academic_credits": "WUT?",
69     "lulz_had": "over 9000",
70     "external_link": "YY",
71     "small_image": "X",
72     "techniques_used": [
73
74     ],
75     "project_name": "",
76     "course_id": " \",
77     "end_date": "2009-09-07",
78     "project_id": 4,
79     "big_image": "XXX"
80 }
81 ]
```

B Feedback - Presentation av Portfolio

Figur 10: Inscannade dokumentet med feed-back från portfolio presentationen.

Detta gjorde jag:	Detta förväntade jag mig skulle hända:	Detta hände:
Klickade i projekt.id och sökte på 2. URL = /list/	Visa projekt med projekt id 2.	Fick förväntat resultat.
Sorterade efter olika söttar i search och klickade på projektet	(info om de olika parametrarna (ex) course_id)	inga info
Sorterade med parameter som inte var "asc" eller "desc"	Felhantering	Internal Server Error
Felaktigt sökfält	Felhantering	—(—
Bör info om fält finnas?		