



**Module: CST 2120 – Web Applications and
Databases**

Coursework 3 – Single Web Page

2021 September Intake

Student Name: Anmol Vishvas

Student Number: M00734701

Contents

1. Website Overview.....	4
2. Website.....	4
2.1. Register Tutor.....	4
2.1.1. Tutor database before registering the tutor	4
2.1.2. Tutor registration	5
2.1.3. Tutor database after registering the tutor	5
2.2. Login Tutor.....	6
2.3. Home Page.....	6
2.4. Register Students	7
2.4.1. Student database before registering students.....	7
2.4.2. Student registration	7
2.4.3. Student database after registering students	7
2.5. View Student Data.....	8
2.5.1. New student data before add subject, level, and grades.....	8
2.5.2. New student data after add subject, level, and grades	9
2.6. Add Subject and Level	9
2.6.1. Student database before adding subject and level	9
2.6.2. Add subject and level.....	10
2.6.3. Student database after adding subject and level	10
2.7. Add Grades	10
2.7.1. Student database before adding grades	10
2.7.2. Add grades.....	11
2.7.3. Student database after adding grades	11
3. Testing.....	12
3.1. HTML Validator	12
3.2. CSS Validator	12

3.3. Selenium Testing	12
3.3.1. Login	12
3.3.2. View student data.....	13
3.3.3. Register Student.....	13
3.3.4. Add subject and level.....	14
3.3.5. Add grades.....	14
3.4. JavaScript Unit tests (Mocha).....	15
3.4.1. Server test output	15
3.4.2. Client test Output.....	15
4. Database Design	15

1. Website Overview

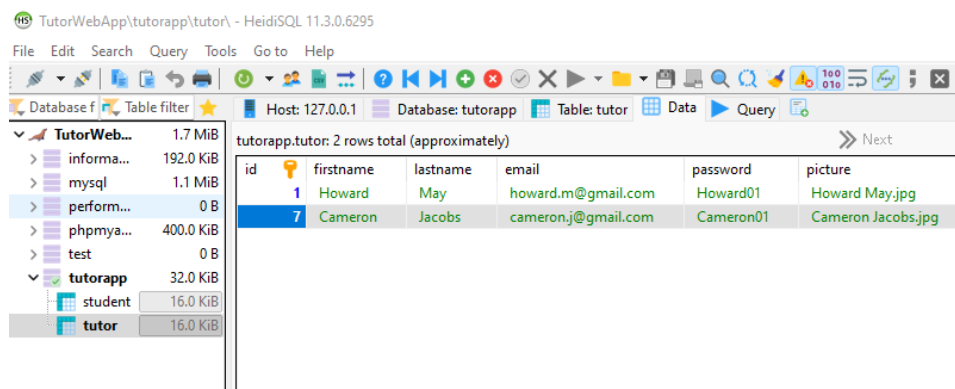
I decided to design a TutorApp in the form of a single page web application. TutorApp will enable the tutor to register a new student using their first name, last name, and e-mail address. The tutor will be allowed to view a specific student's data such as his or her first name, level, subject on which he or she was enrolled and his or her grades in each subject by just entering the student's surname.

The second feature in this website, is that the tutor is allowed to add more subjects and level for a student and lastly, he can also add grades. We will use SQL to store data. We will have two tables, tutor table that will store the tutor's id, first name, last name, e-mail address and password that he or she will use to login on the website. The second table is the student table with the student's id, first name, last name, e-mail address, level, and subject.

2. Website

2.1. Register Tutor

2.1.1. Tutor database before registering the tutor



The screenshot shows the HeidiSQL interface. On the left, a tree view displays the database structure, including 'TutorWebApp' and 'tutorapp' databases. The 'tutor' table is selected. The main window shows the data for the 'tutor' table, which contains two rows. The columns are: id, firstname, lastname, email, password, and picture.

id	firstname	lastname	email	password	picture
1	Howard	May	howard.m@gmail.com	Howard01	Howard May.jpg
7	Cameron	Jacobs	cameron.j@gmail.com	Cameron01	Cameron Jacobs.jpg

As we can see above, we have only two tutor details stored into the database.

2.1.2. Tutor registration

Navigation bar

Registration form

Footer of the webpage

2.1.3. Tutor database after registering the tutor

TutorWebApp\tutorapp\tutor\ - HeidiSQL 11.3.0.6295

File Edit Search Query Tools Go to Help

Database f Table filter Host: 127.0.0.1 Database: tutorapp Table: tutor Data Query

▼ TutorWeb... 1.7 MiB

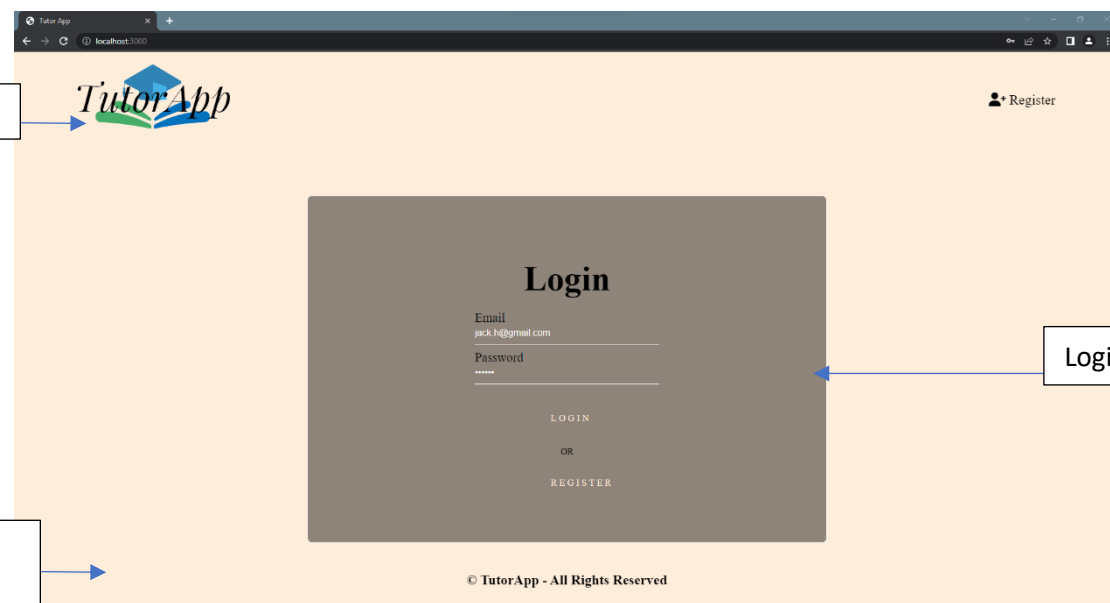
- > informa... 192.0 KiB
- > mysql 1.1 MiB
- > perform... 0 B
- > phpmya... 400.0 KiB
- > test 0 B
- ▼ tutorapp 32.0 KiB
 - student 16.0 KiB
 - tutor 16.0 KiB

tutorapp.tutor: 3 rows total (approximately)

id	firstname	lastname	email	password	picture
1	Howard	May	howard.m@gmail.com	Howard01	Howard May.jpg
7	Cameron	Jacobs	cameron.j@gmail.com	Cameron01	Cameron Jacobs.jpg
14	Jack	Harris	jack.h@gmail.com	Jack01	Jack Harris.jpg

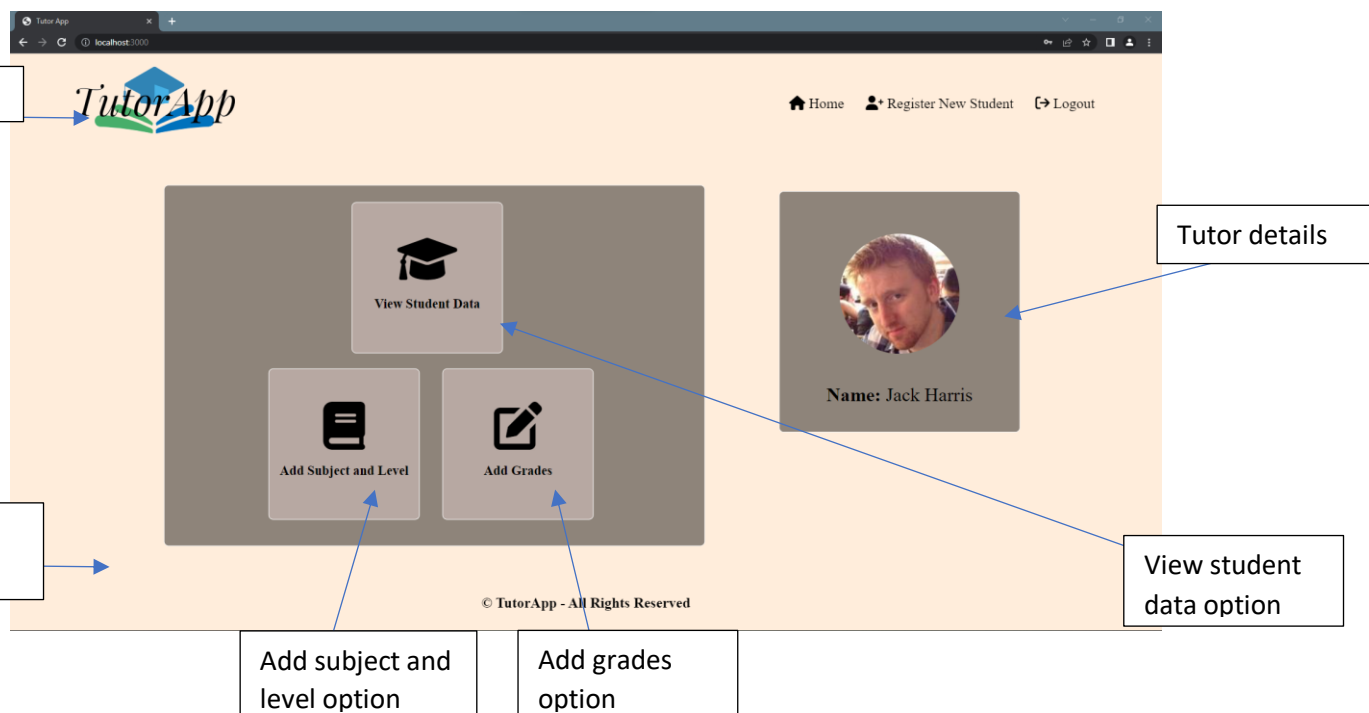
As we can see above, we have now three tutor details stored into the database.

2.2. Login Tutor



Once the user will login, if the credential matches to the one stored into the database, the user will be redirected to the home page.

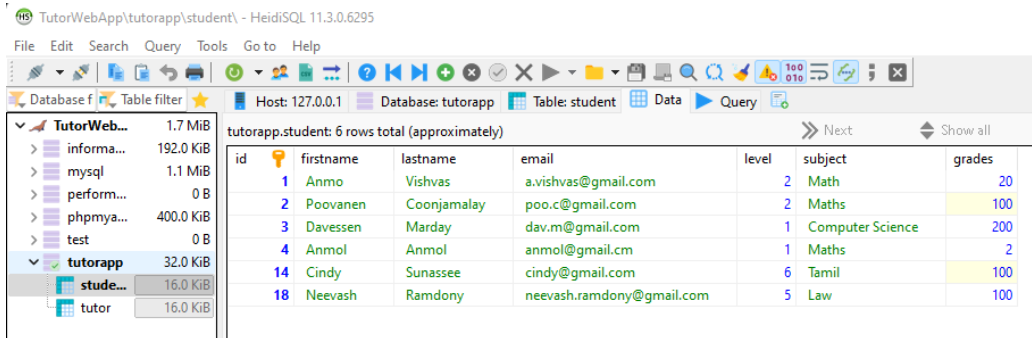
2.3. Home Page



Once the user logged in, he or she will be redirected to this page. From this page, the tutor will be able to “register a new student”, “view student data”, “add subject and level” and “add grades”.

2.4. Register Students

2.4.1. Student database before registering students



TutorWebApp\tutorapp\student\ - HeidiSQL 11.3.0.6295

File Edit Search Query Tools Go to Help

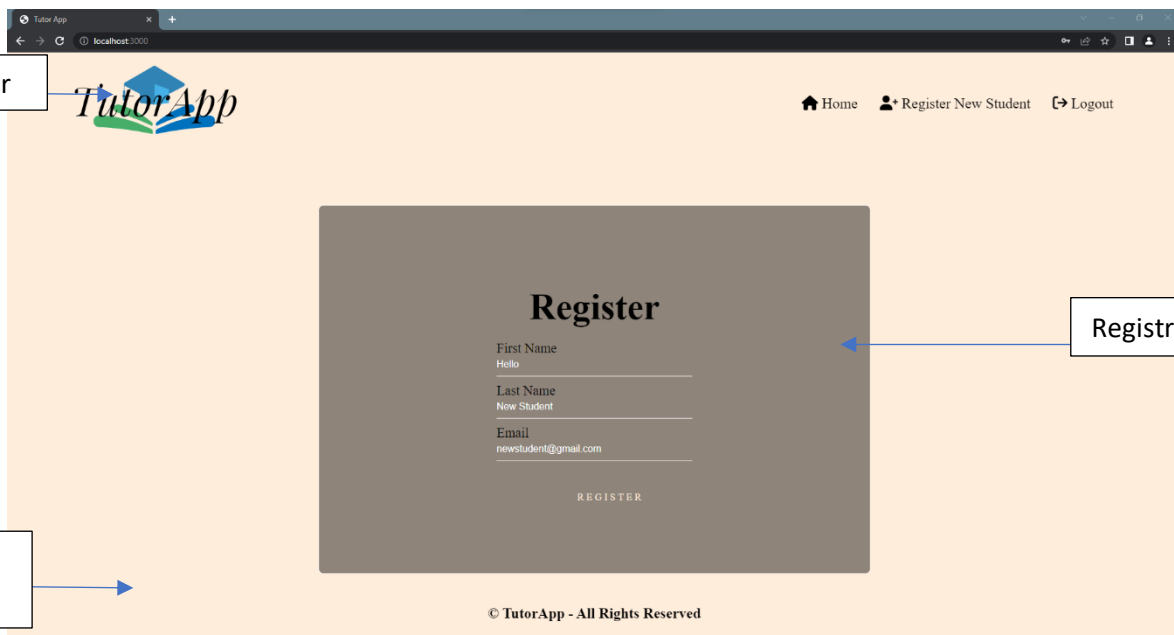
Database: tutorapp Table: student Data Query

tutorapp.student: 6 rows total (approximately)

id	firstname	lastname	email	level	subject	grades
1	Anmo	Vishvas	a.vishvas@gmail.com	2	Math	20
2	Poovanen	Coonjamalay	poo.c@gmail.com	2	Maths	100
3	Davessen	Marday	dav.m@gmail.com	1	Computer Science	200
4	Anmol	Anmol	anmol@gmail.cm	1	Maths	2
14	Cindy	Sunassee	cindy@gmail.com	6	Tamil	100
18	Neevash	Ramdony	neevash.ramdony@gmail.com	5	Law	100

As we can see above, we have only six student details stored into database.

2.4.2. Student registration



Navigation bar

TutorApp

Home Register New Student Logout

Register

First Name
Hello

Last Name
Now Student

Email
newstudent@gmail.com

REGISTRATION FORM

Registration form

Footer of the webpage

© TutorApp - All Rights Reserved

2.4.3. Student database after registering students

TutorWebApp\tutorapp\student\ - HeidiSQL 11.3.0.6295

File Edit Search Query Tools Go to Help

Database f Table filter Host: 127.0.0.1 Database: tutorapp Table: student Data Query

tutorapp.student: 7 rows total (approximately) >> Next Show all

id	firstname	lastname	email	level	subject	grades
1	Anmo	Vishvas	a.vishvas@gmail.com	2	Math	20
2	Poovanen	Coonjamalay	poo.c@gmail.com	2	Maths	100
3	Davessen	Marday	dav.m@gmail.com	1	Computer Science	200
4	Anmol	Anmol	anmol@gmail.cm	1	Maths	2
14	Cindy	Sunasse	cindy@gmail.com	6	Tamil	100
18	Neevash	Ramdony	neevash.ramdony@gmail.com	5	Law	100
19	Hello	New Student	newstudent@gmail.com	0		0

As we can see above, a new user has been stored into the database. Initially, the new student will have a 0 level and grade 0.

2.5. View Student Data

2.5.1. New student data before add subject, level, and grades

The screenshot shows the TutorApp web interface. A navigation bar at the top contains the TutorApp logo and links for Home, Register New Student, and Logout. The main content area features a search form titled "Show Data of a Student" with input fields for Surname and New Student, and a SEARCH button. Below the search form, a section displays student details: FirstName: Hello, Level: Student level not updated yet, and Subject: No subject assigned to the student. A footer at the bottom reads "© TutorApp - All Rights Reserved".

Navigation bar

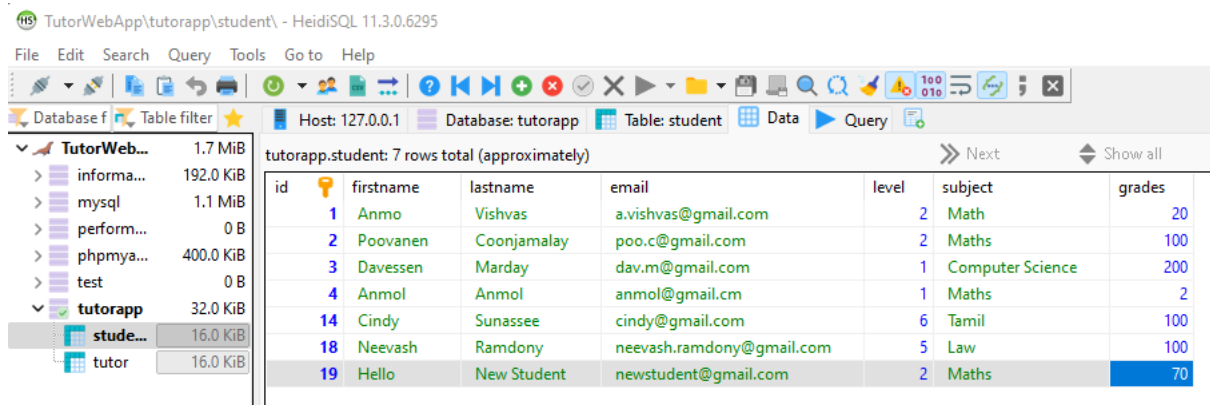
Search form

Section where student details will be displayed once "search" button will be pressed.

Footer

Since the student have just been registered, he or she was not yet assigned to a level and subject. Therefore, we are getting this message.

2.5.2. New student data after add subject, level, and grades



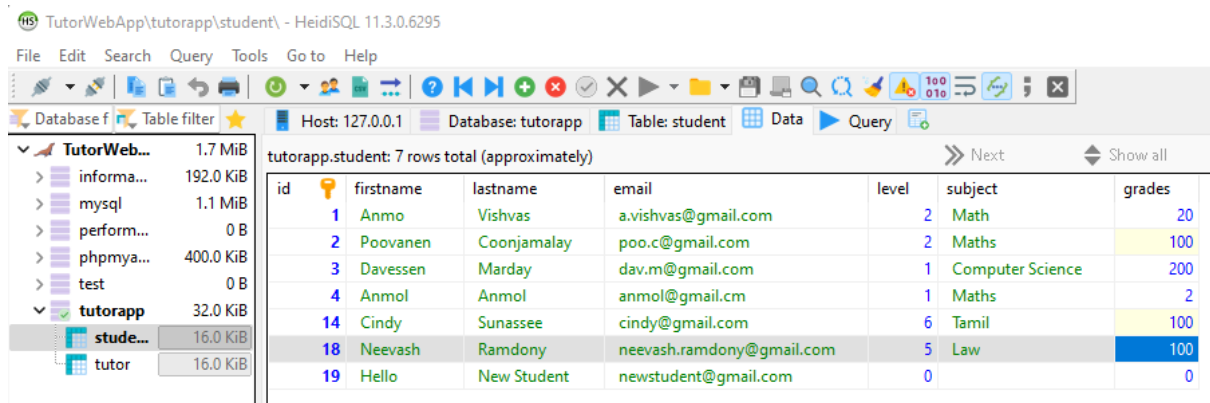
tutorapp.student: 7 rows total (approximately)

id	firstname	lastname	email	level	subject	grades
1	Anmo	Vishvas	a.vishvas@gmail.com	2	Math	20
2	Poovanen	Coonjamalay	poo.c@gmail.com	2	Maths	100
3	Davessen	Marday	dav.m@gmail.com	1	Computer Science	200
4	Anmol	Anmol	anmol@gmail.cm	1	Maths	2
14	Cindy	Sunassee	cindy@gmail.com	6	Tamil	100
18	Neevash	Ramdony	neevash.ramdony@gmail.com	5	Law	100
19	Hello	New Student	newstudent@gmail.com	2	Maths	70

As we can see above, the database has been updated with the data inputted.

2.6. Add Subject and Level

2.6.1. Student database before adding subject and level



tutorapp.student: 7 rows total (approximately)

id	firstname	lastname	email	level	subject	grades
1	Anmo	Vishvas	a.vishvas@gmail.com	2	Math	20
2	Poovanen	Coonjamalay	poo.c@gmail.com	2	Maths	100
3	Davessen	Marday	dav.m@gmail.com	1	Computer Science	200
4	Anmol	Anmol	anmol@gmail.cm	1	Maths	2
14	Cindy	Sunassee	cindy@gmail.com	6	Tamil	100
18	Neevash	Ramdony	neevash.ramdony@gmail.com	5	Law	100
19	Hello	New Student	newstudent@gmail.com	0		0

As we can see above, Hello student has a 0 level and no subject.

2.6.2. Add subject and level

Navigation bar

Footer

Add subject and level form

Add Subject and Level

Surname
New Student

Level
2

Subject
Maths

SEARCH

DONE

© TutorApp - All Rights Reserved

2.6.3. Student database after adding subject and level

TutorWebApp\tutorapp\student\ - HeidiSQL 11.3.0.6295

File Edit Search Query Tools Go to Help

Database f Table filter Host: 127.0.0.1 Database: tutorapp Table: student Data Query

tutorapp.student: 7 rows total (approximately)

id	firstname	lastname	email	level	subject	grades
1	Anmo	Vishvas	a.vishvas@gmail.com	2	Math	20
2	Poovanen	Coonjamalay	poo.c@gmail.com	2	Maths	100
3	Davessen	Marday	dav.m@gmail.com	1	Computer Science	200
4	Anmol	Anmol	anmol@gmail.cm	1	Maths	2
14	Cindy	Sunassee	cindy@gmail.com	6	Tamil	100
18	Neevash	Ramdony	neevash.ramdony@gmail.com	5	Law	100
19	Hello	New Student	newstudent@gmail.com	2	Maths	0

As we can see above, the database has been updated with the data inputted.

2.7. Add Grades

2.7.1. Student database before adding grades

TutorWebApp\tutorapp\student\ - HeidiSQL 11.3.0.6295

File Edit Search Query Tools Go to Help

Database f Table filter Host: 127.0.0.1 Database: tutorapp Table: student Data Query

tutorapp.student: 7 rows total (approximately)

id	firstname	lastname	email	level	subject	grades
1	Anmo	Vishvas	a.vishvas@gmail.com	2	Math	20
2	Poovanen	Coonjamalay	poo.c@gmail.com	2	Maths	100
3	Davessen	Marday	dav.m@gmail.com	1	Computer Science	200
4	Anmol	Anmol	anmol@gmail.cm	1	Maths	2
14	Cindy	Sunassee	cindy@gmail.com	6	Tamil	100
18	Neevash	Ramdony	neevash.ramdony@gmail.com	5	Law	100
19	Hello	New Student	newstudent@gmail.com	2	Maths	0

As we can see above, Hello student have 0 as grade.

2.7.2. Add grades

Navigation Bar

Add grades form

Footer

2.7.3. Student database after adding grades

TutorWebApp\tutorapp\student\ - HeidiSQL 11.3.0.6295

File Edit Search Query Tools Go to Help

Database f Table filter Host: 127.0.0.1 Database: tutorapp Table: student Data Query

tutorapp.student: 7 rows total (approximately)

id	firstname	lastname	email	level	subject	grades
1	Anmo	Vishvas	a.vishvas@gmail.com	2	Math	20
2	Poovanen	Coonjamalay	poo.c@gmail.com	2	Maths	100
3	Davessen	Marday	dav.m@gmail.com	1	Computer Science	200
4	Anmol	Anmol	anmol@gmail.cm	1	Maths	2
14	Cindy	Sunassee	cindy@gmail.com	6	Tamil	100
18	Neevash	Ramdony	neevash.ramdony@gmail.com	5	Law	100
19	Hello	New Student	newstudent@gmail.com	2	Maths	70

As we can see above, the database has been updated with the data inputted.

3. Testing

3.1. HTML Validator

Nu Html Checker

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

Showing results for uploaded file index.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Browse...](#) No file selected.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.


[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 16 milliseconds.

3.2. CSS Validator



The W3C CSS Validation Service

W3C CSS Validator results for style.css (CSS level 3 + SVG)

W3C CSS Validator results for style.css (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as [CSS level 3 + SVG](#) !

3.3. Selenium Testing

3.3.1. Login

Project: Test 1

Executing: ~

✓ Login Test*

http://localhost:3000

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1936x1048	
3	✓ click	id=tutorial	
4	✓ type	id=tutorial	howard.m@gmail.com
5	✓ type	id=tutorialpass	Howard01
6	✓ click	linkText=LOGIN	
7	✓ assert value	id=tutorial	howard.m@gmail.com
8	✓ click	linkText=Logout	

Command

Target

Value

Description

Runs: 1 - Failures: 0

Log	Reference
1. open on / OK	12:56:19
2. setWindowSize on 1936x1048 OK	12:56:19
3. click on id=tutorial OK	12:56:19
4. type on id=tutorial with value howard.m@gmail.com OK	12:56:20
5. type on id=tutorialpass with value Howard01 OK	12:56:20
6. click on linkText=LOGIN OK	12:56:20
7. assertValue on id=tutorial with value howard.m@gmail.com OK	12:56:20
8. click on linkText=Logout OK	12:56:21
Login Test* completed successfully	

3.3.2. View student data

Extension: (Selenium IDE) - Selenium IDE - Test 2* - Mozilla Firefox

Project: Test 2*

Executing -

http://localhost:3000

Command	Target	Value
1. ✓ open	/	
2. ✓ set window size	1936x1048	
3. ✓ click	css=home_option1 > p	
4. ✓ click	id=surname	
5. ✓ type	id=surname	Vishvas
6. ✓ click	css=user-box > a	
7. ✓ assert text	css=details>span	Anmo

Runs: 1 Failures: 0

Log Reference

Running 'View student data'

1. open on / OK	14:43:12
2. setWindowSize on 1936x1048 OK	14:43:13
3. click on css=home_option1 > p OK	14:43:13
4. click on id=surname OK	14:43:14
5. type on id=surname with value Vishvas OK	14:43:14
6. click on css=user-box > a OK	14:43:14
7. assertText on css=details>span with value Anmo OK	14:43:14
'View student data' completed successfully	14:43:14

3.3.3. Register Student

Extension: (Selenium IDE) - Selenium IDE - Test 3* - Mozilla Firefox

Project: Test 3*

Executing -

http://localhost:3000

Command	Target	Value
1. ✓ open	/	
2. ✓ set window size	1936x1048	
3. ✓ click	css=#HomeSection a:nth-child(2)	
4. ✓ click	name=studentname	
5. ✓ type	name=studentname	Ayush
6. ✓ click	name=studentname	
7. ✓ type	name=studentname	Vishvas
8. ✓ click	name=studentEmail	
9. ✓ type	name=studentEmail	ayush@gmail.com
10. ✓ assert value	name=studentname	Vishvas
11. ✓ click	css=form > a:nth-child(4)	

Runs: 1 Failures: 0

Log Reference

Running 'Register Student'

4. click on name=studentname OK	14:57:03
5. type on name=studentname with value Ayush OK	14:57:03
6. click on name=studentname OK	14:57:04
7. type on name=studentname with value Vishvas OK	14:57:04
8. click on name=studentEmail OK	14:57:04
9. type on name=studentEmail with value ayush@gmail.com OK	14:57:04
10. assertValue on name=studentname with value Vishvas OK	14:57:04
11. click on css=form > a:nth-child(4) OK	14:57:04
'Register Student' completed successfully	14:57:04

3.3.4. Add subject and level

Extensions: (Selenium IDE) - Selenium IDE - Test 4* — Mozilla Firefox

Project: Test 4*

Tests +

Search tests...

http://localhost:3000

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1936x1048	
3	✓ click	css=fa-book	
4	✓ click	id=studentSurname	
5	✓ type	id=studentSurname	Vishvas
6	✓ click	css=login-box a:nth-child(2)	
7	✓ click	id=studentSubject	
8	✓ type	id=studentSubject	Math
9	✓ assert value	id=studentSubject	Math
10	✓ click	css=#detailsStudentLevel > form > a	

Command #

Target

Value

Description

Log Reference

3. click on css=fa-book OK 15:21:51

4. click on id=studentSurname OK 15:21:52

5. type on id=studentSurname with value Vishvas OK 15:21:52

6. click on css=login-box a:nth-child(2) OK 15:21:52

7. click on id=studentSubject OK 15:21:52

8. type on id=studentSubject with value Math OK 15:21:52

9. assertValue on id=studentSubject with value Math OK 15:21:53

10. click on css=#detailsStudentLevel > form > a OK 15:21:53

'Add Subject and Level' completed successfully 15:21:53

3.3.5. Add grades

Extensions: (Selenium IDE) - Selenium IDE - Test 5* — Mozilla Firefox

Project: Test 5*

Executing -

✓ Add Grades*

http://localhost:3000

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1936x1048	
3	✓ click	css=home_option3	
4	✓ click	id=SurnameStudent	
5	✓ type	id=SurnameStudent	Vishvas
6	✓ click	css=grade-box > form > a	
7	✓ click	id=grade	
8	✓ type	id=grade	20
9	✓ assert value	css=inputSubject	Math
10	✓ click	css=grade > a	

Command #

Target

Value

Description

Runs: 1 Failures: 0

Log Reference

3. click on css=home_option3 OK 15:33:25

4. click on id=SurnameStudent OK 15:33:25

5. type on id=SurnameStudent with value Vishvas OK 15:33:26

6. click on css=grade-box > form > a OK 15:33:26

7. click on id=grade OK 15:33:26

8. type on id=grade with value 20 OK 15:33:26

9. assertValue on css=inputSubject with value Math OK 15:33:26

10. click on css=grade > a OK 15:33:26

'Add Grades' completed successfully 15:33:26

3.4. JavaScript Unit tests (Mocha)

3.4.1. Server test output

```
PS C:\www> npm test

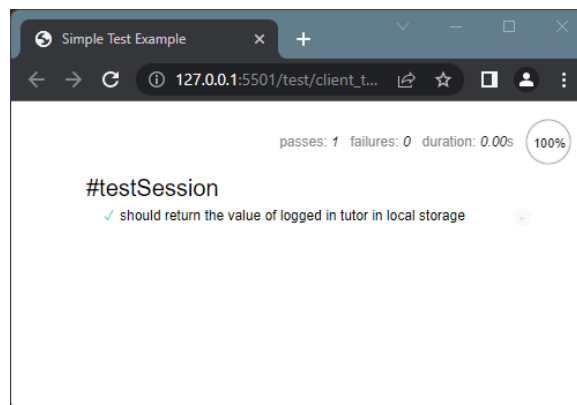
> tutorapp@1.0.0 test
> mocha

Database
  #getAllStudents
    ✓ should return all of the students in the database
  #addTutor
    ✓ should add a tutor to the database
  #addStudents
    ✓ should add a student to the database

Web Service
  /GET tutor
    ✓ should GET all the student

4 passing (82ms)
```

3.4.2. Client test Output



4. Database Design

