Web Report (coursework 2)

Web platform development 2

Abstract

This report includes the link design of the application, a description of the persistence mechanism, the functionality of the application and test results. Finally, a description and justification of the security features implemented including security measures that may not have been implemented but could have been.  
  
“I declare that all work submitted for this coursework is the work of Arron Fairley alone unless stated otherwise”

Fairley, Arron

Contents

[1.1 The Link Design 2](#_Toc38048597)

[2.1 The Persistence Mechanism 5](#_Toc38048598)

[2.2 The Database Schema 6](#_Toc38048599)

[2.3 The Data Access Layer 7](#_Toc38048600)

[3.1 Functionality Of The Application 8](#_Toc38048601)

[3.1.1 Moscow Analysis 8](#_Toc38048602)

[3.2 Test Reports 8](#_Toc38048603)

[3.2.1 Functionality Test 8](#_Toc38048604)

[3.3 Summary Test Results 10](#_Toc38048605)

[4.1 Application Security 10](#_Toc38048606)

[4.2 Possible future security implementation 10](#_Toc38048607)

[4.2.1 Complex passwords 10](#_Toc38048608)

[4.2.2 SSL Service 11](#_Toc38048609)

[4.2.3 Web Hosting 11](#_Toc38048610)

[4.2.4 Website firewall 11](#_Toc38048611)

[4.2.5 Two Factor Authentication 11](#_Toc38048612)

Figure 1- Link design 4

Figure 2 - User ID being used to link coursework 6

Figure 3 - Database Schema 6

Figure 4 - Moscow analysis 8

Figure 5 - Functionality Test 10

# 1.1 The Link Design

As the user first connects to the website, the User database and the coursework database are both loaded up in the background. The user is then met with the login page. At this point the user then has two options. Either the user can choose to log in if they have registered to the website before, by using their email address and password that was used to register to the website. The website will connect to the user database and compare the inputted email and password against the saved database. If the two match up, the user will be logged into the system.

The other option the user has at this point is to register onto the website. If this link is selected, the user will be taken to another page where the user can register to log onto the website. This will be established by the user entering in their name, email and then creating a password. Once the submit button is selected, this information will be saved into the user database. After registration is completed the user will be taken back to the login page in order for the user to enter in their email and password. After the checks are made, the user will be logged in and taken to the home page.

Once the user is in the home page, they will be met with a customized message welcoming them to the website. This message will include the phrase “hi (name), welcome to your coursework scheduling dashboard” The name will be read from the logged in user in the database. The user will be able to log out at this point. If the user logs out, the session expires and the user is taken to the login page. If the user decides to view the coursework. The user will be taken to the view coursework page. At this point the Coursework database is referenced in relation to the user connected. If there is any previously entered coursework’s into the users profile, the data is read and displayed on the website showing the coursework title, its description and its milestones involved In the coursework. The entry is also listed along with a due completion date.

If there is no existing coursework’s for the current logged in user. The user will be shown a blank page with the option to add a new coursework. If this option is selected the user will be taken to the add/edit coursework page. In this webpage, the user will be prompted to add in the relevant information for the coursework. The information required for the coursework to be added is as follows. The coursework Title, the description of the coursework, the milestones involved and the due date of the coursework to be completed. Only after this information is entered will the website allow the user to carry on adding the coursework. Once all the prerequisites have been met and the submit button pressed. The coursework details will be added into the coursework database and the entry will be linked to the current user. This will allow for the coursework to be retrieved when the user logs in.

If there is existing coursework the user will be able to edit the coursework’s by clicking on the add/edit button. When taken to the add/edit coursework page, the user will be able to select elements of the coursework that they want to change and change them appropriately. Once the submit button is selected the newly changed information existing information on the database will be amended with the newly entered information.

Below ([see Figure 1](#_The_Link_Design)) is a Link design diagram showing the overall flow of the website. This is a useful addition as is shows graphically how the website links together.

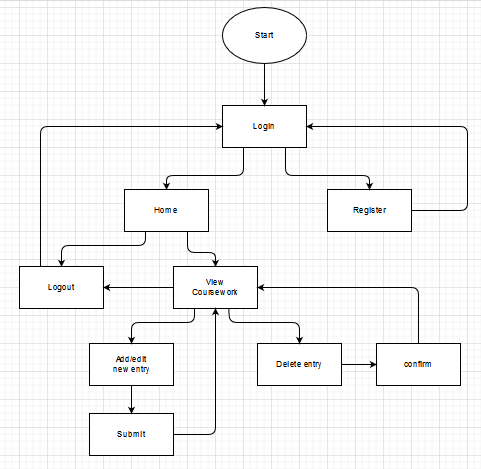


Figure 1- Link design

The above link design was intended to make it easy for the user to navigate between pages. Once the user is logged in all functionalities are available to the user from the first page that loads to them. The links are concise and are easy to see making navigation simple.

# 2.1 The Persistence Mechanism

The persistence mechanism used in the creation was a module called NEDB. This module is part of the node JS package. The NEDB module allows for the storage of all the users register information that is inputted into the register page. The information that the user inputs into the online form is saved into the root file of where the code is being executed. This is the default location that the database will save to if not coded to save elsewhere. The default folder was kept as it meant that the database will be kept in the same folder as the application. This makes it more suitable for transportation. If the database save location was to be changed to somewhere on its origin machine and the application was to be moved to another machine. The application may not be able to find the location of where the database is supposed to be read or saved to.

NEDB was used as it is a lightweight database. This means that the resource impact of the database is as small as possible. This is useful as it decreases loading times and creates a smoother user experience. There is other large-scale database API or extensions that could be used, such as Mongodb. However, these types of database are unnecessary as the website is only small scale and large storage locations or more advanced features are not needed for this website. Mongo DB extends from NEDB and allows for some interesting features such as IP connectivity and large save files for the use of many users. This does however slow the loading times of the application.

The code developed using NEDB allows for the data in the database to be read and written to by a logged in user. It also allows for that data to be manipulated by the user logged in. This functionality is the core functions of the website to be developed such as adding / editing the coursework and the function that allows the user to remove coursework. The relevant data relating to the user is also read and displayed on the webpage for the user in the form of a list. This is another use of the NEDB working. The registration for the user is also directly connected to the use of NEDB as it allows for inputted information to be saved to the user database.

## 2.2 The Database Schema

The User DB and the Coursework DB are linked together as it allows the user to view the saved information that is related to the users register information. When the User is created a unique coursework ID is automatically input into the users register information. This unique number is then used for when the user creates a new input to the coursework database ([see Figure 2](#_2.2_The_Database)) and ([Figure 3](#_2.2_The_Database))

User ID being used to link to coursework

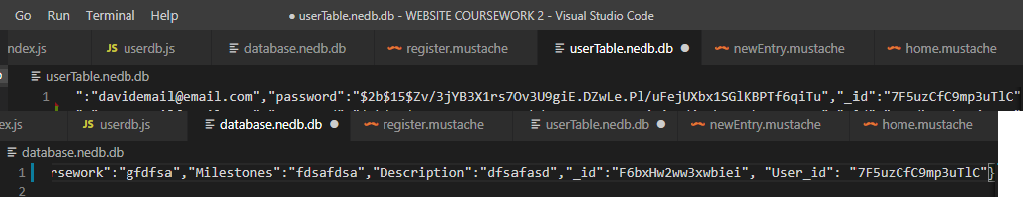


Figure 2 - User ID being used to link coursework

Database Schema

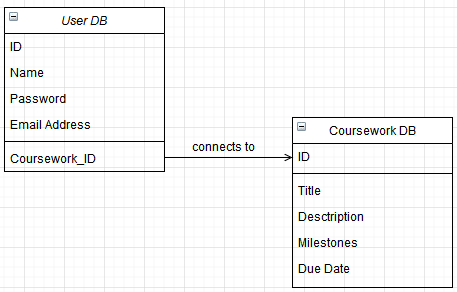


Figure 3 - Database Schema

## 2.3 The Data Access Layer

This layer of the application allows for the main functionality of the website to be realized. Models were created for each of the databases which included the implementation of these features. The User DB has a constructor that loads the existing database into the website. If one does not exist a new one is created in the root folder. The function “All” is also executed. This function loads up all the existing users into the website. This allows for the user to login to the website using the email and password previously created. The function checks the email and password against the saved data and logs the user in. The function “Add” is used if the user is not registered to the website. This allows the user to input the data required into the database allowing the data to be used to create an account.

The coursework database also has this constructor that allows the coursework data to be loaded into the website. If no database exists, the constructor will create a new database for the coursework to be added into. The coursework database model also has an “All” function. This returns the list of available coursework’s with all the data available for each coursework. This is useful for the listing of the coursework’s on the main page. The add function takes the inputted data from the user and saves the data as an “entry”. This entry is what is recalled when the “All” function is called.

# 3.1 Functionality Of The Application

A Moscow analysis was carried out on the description of the application. By doing the Moscow analysis the prioritized functions were set out along with other optional functions that could be implemented with the first version of the website. The Moscow analysis is below ([Table](#_3.1.1_Moscow_analysis) 1).

### 3.1.1 Moscow Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Must have** | **Should have** | **Could have** | **Won’t have** |
| Add Coursework | Log in feature | Module adding (grouping) | **N/A** |
| Remove Coursework | Secure password save | Sort coursework filter | **N/A** |
| Edit Coursework |  |  | **N/A** |
| Shareable link |  |  | **N/A** |

Figure 4 - Moscow analysis

## 3.2 Test Reports

In order to establish if the specifications were met in the implementation of the website. Tests had to be commenced and documented on the running website. One of which test reports were a functionality test. ([Table 2](#_3.2_Test_Reports))

### 3.2.1 Functionality Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test | Expectation | Pass/Fail | Screenshot |
|  | Click on the add new entry on the view coursework page | The user will be taken to the add new entry page | PASS |  |
|  | Fill in the form and click the submit button | The new coursework will be added and the user taken to the view coursework page | PASS |  |
|  | Click remove coursework on the view coursework page | The user will be met with a confirmation | PASS |  |
|  | Click okay in the confirmation page | The user will be taken back to the view coursework’s page with the deleted coursework removed | PASS |  |
|  | Click the edit coursework button on the view coursework page | Coursework should be listed with content to be edited | PASS |  |
|  | Enter the appropriate data to be edited and click submit | The coursework will be edited and then the user taken back to the view coursework page | PASS |  |
|  | Paste the provided link to share course information along with the name of who sent it | The link provided displays | PASS |  |
|  | Click the show me button on the shared link | Shows the shared coursework | PASS |  |
|  | log in with a registered username and password clicking login | The user will be logged into the home page (customized with name) | PASS |  |
|  | Click on the coursework button on the home page | The user will be taken to the view coursework page where coursework’s will be listed along with the option buttons | PASS |  |
|  | Click “register” on the login page | User is taken to the register page | PASS |  |
|  | Enter details then click register | The user will be taken to the login page to login | PASS |  |
|  | Do not enter information and click register | User should be prompted with a message to fill the field in | PASS |  |
|  | Click on the log out button | The user will be logged out and taken to the login page | PASS |  |

Figure 5 - Functionality Test

# 3.3 Summary Test Results

Upon completing the functionality test above it was apparent that all functionality does indeed work as expected. The website works well as a completed product. Overall progress seems to be okay. Perhaps in the future, other functionalities could be implemented. The webpage is also in need of some design. This will be completed before final upload and changes will be pushed to GitHub.

# 3.4 No peer Review

Seeing as this was an individual project there was no other team members to analyze the code as it was being implemented. This caused issues in testing the implementation as the tests carried out as an individual may not be as thorough as a project that was created in a team environment. As an individual project it is more likely that some bugs and/or features may have been missed or not work as intended. As a result, the project was created with research from previous labs and online research. The tests that were carried out above were designed to realize the working functionality of the application.

# 4.1 Application Security

The main security that was involved in this development was the use of Bcrypt. Bcrypt is a hashing function that is used on the password that the user inputs into the database when they register. The password is hashed and salted. The salt is the random characters used to hash the password. This makes he users password impossible to guess even if the database was to be breached. The password that would be recovered would just be a mess of characters.

Usernames and passwords are also compared against the information in the database when entered into the website. This is to ensure the person logging into the website is the intended register user.

There may be some security issues in the application that could be fixed. One of which concerns is the shareable link that is on the page. If anyone was to gain knowledge of this link it could be possible to use it to gain access into the other users coursework lists as the page is designed to retrieve all the coursework’s entered and return the relevant users lists to the page.

## 4.2 Possible future security implementation

### 4.2.1 Complex passwords

The password that is to be entered into the registration could be improved. It could be possible to prompt the user to create a stronger password. At the moment there is no level of the password complexity being enforced. If the password was to contain symbols and at least a capital letter and numbers, the password would be harder to break. This could be implemented into the website asking to create a password that is more complex.

### 4.2.2 SSL Service

A service that could be used in the future could be an SSL service such as comodo. This allows the website and all of its data involved in it to be encrypted. Not just the password. This would mean that the user could see that the security of the application is more advanced as it shows on the address bar as https and a verification seal that is placed on the web page.

### 4.2.3 Web Hosting

A certified web host would also be a good choice. This would be beneficial to the site as it would allow for the site to benefit from faster loading times. The information handled through the website would also be more secure as these hosting services have more advanced security than just the website alone. Also the hosting service will have a team that will be monitoring all the websites associated with the application. If there was to be any issues with the website, the host would be able to identify and solve the issue quickly and professionally. This would minimize downtime for the users if something was to go wrong.

A lot of web hosting services also allow for automatic backups and updates to security. This would mean that as new security features were released the website would stay up to date and more importantly, stay secure.

### 4.2.4 Website firewall

It is also possible to install a firewall into the application. There are services such as secure security that are firewalls to protect the website from ddos hackers. These hacker basically ping your website (asking for the connection time and information) so rapidly it makes the server overload and crash. The fire wall will stop this from happening. It also scans the websites file structures to ensure the integrity of the files are still normal on a regular basis.

### 4.2.5 Two Factor Authentication

Google has a two factor authentication tool. This can be plugged into the website to ensure another level of security. This feature would allow the user to get a code texted or emailed to their phone or pc. With this code it would authenticate that the user is indeed who they say they are before allowed access into the website.