Coursework 2: Social Networking Website

1. Summary

Version: 1.2

Individual project

Weighting: 50% of overall mark

Deadline for project proposal: 16:00 Friday 26th January 2024 (end of Week 15)

Deadline for front end and basic web service: 16:00 Friday 23rd February 2024 (end of Week 19)

Deadline for final submission: 16:00 Friday 12th April 2024 (end of Week 24)

2. Key Points

- You select a social networking website that you want to build! It could be a website for photo sharing, file sharing, recipe sharing, sports fans, etc. Come up with your own ideas!
- **The front end of the website must have a single HTML page**. JavaScript will change parts of the page in response to user input.
- The front end must be written in HTML, CSS and JavaScript. You cannot write an app.
- The back end must be implemented as a web service. This must be implemented using JavaScript running on Node.js. **No marks are available for functionality that is implemented in PHP**.
- The path of your web service must start with your student ID. For example, http://localhost:8080/M0012345/. **Zero marks will be awarded for web services whose paths do not start with your student ID**.
- The front end of your website must communicate with the back end using AJAX.
- After the single HTML page has loaded only JSON formatted data can be exchanged between client and server.
- jQuery, Bootstrap and Foundation are the *only* third-party libraries permitted for the *front end* of this project. Other front-end libraries can only be used with the explicit permission of your laboratory teacher or module leader. **React, Ember, Angular and Vue are banned**. Functionality implemented with banned third party libraries will receive zero marks.
- There are no restrictions on the Node modules that you can use for the *back end* of this project.
- You must use MongoDB to store data for the website.
- Cross-browser compatibility is not required – you only need to demonstrate your website on a single browser.
- Marks are available for code quality, testing and for the front end of the website.
- Your final report should include a discussion of the security, privacy and legal issues that affect your website.
- The final submission of your project must include a video demonstration that shows all the functionality.

3. Technology

Coursework 2 must be implemented using the following technology:

- The front end must be written in HTML, CSS and JavaScript. You cannot use JavaFX or write an app.
- The front end must consist of a *single HTML page*. JavaScript will be used to change the contents of this page.
- The single HTML page can be sent from the server *once* when the website loads. *Once the single page has loaded no HTML can be sent between client and server*.

- The server-side functionality must be a web service that is implemented as using JavaScript running on Node.js.
- Your web service path must start with your student ID. For example, http://localhost:8080/M0012345/.
- JavaScript on the front end must communicate with the back end using AJAX.
- Client and server JavaScript can only exchange data in JSON format. Upload of images or files is the only exception to this.
- jQuery, Bootstrap and Foundation are the *only* third-party libraries permitted for the *front end* of this project. Other front-end libraries can only be used with the explicit permission of your laboratory teacher or module leader. *React, Ember, Angular and Vue are banned*.
- There are no restrictions on the Node modules that you can use for the *back end* of this project.
- Data must be stored in MongoDB.
- At least one document containing your name, student ID and student email address must be stored by the basic web service in the second stage of the project.

Marks will only be awarded for websites that follow these rules.

4. What Needs to be Submitted:

4.1 Project Proposal (16:00 Friday 26th January 2024)

Submit the following:

1. *Proposal.* This should contain:
   - Brief description of proposed website.
   - Wireframe sketches of website or screenshots of early prototypes.
2. *MongoDB database design*. List the collections and give examples of MongoDB documents in each collection. Example documents can be screenshots, JSON pasted into your proposal or JSON files that are included in a zip file with the proposal. A database dump of an implementation of the design can also be included with the proposal.

The proposal document must be in Word or PDF format.

We will use the proposal to give you feedback about your idea and help you realize it in the time available. You can reuse material from the project proposal in the final project submission.

Submit the project proposal using the link in the Coursework 2 section of the course website.

The project proposal is worth 10% of the marks for Coursework 2.

4.2 Front End and Basic Web Service (Deadline: 16:00 Friday 23rd February 2024)

Submit a zip file that contains:

1. *Front end and basic web service documentation*. This should contain:
   - Screenshots of front end of website.
   - Screenshots of Postman showing GET and POST/PUT functionality working.
   - Screenshots of MongoDB showing stored data. At least one of the stored documents must contain your name, student ID and student email address.

   This must be in Word or PDF format.
2. *Front end website code*. HTML, CSS, JavaScript and images for the front end of the website.
3. *Basic web service code*. JavaScript code for your basic web service. **Please do not include the node_modules folder in your submission. It could make it too large to upload!**

The front end and basic web service submission will be marked online. It is worth 30% of the marks for Coursework 2.

4.3 Final Submission (Deadline: 16:00 Friday 12th April 2024)

Submit a zip file that contains:

1. *Project report*. This should contain:
   - Clear description of final project.
   - Discussion of the security, privacy and legal issues that affect the website.
   - Screenshots showing the results of HTML and CSS validation.
   - Screenshots showing the results of JavaScript unit tests.
   - Screenshots of front end of the website.
   - Screenshots of Postman showing all the functionality of your final web service.

   This document must be in Word or PDF format.
2. *Front end code*. HTML, CSS, JavaScript and images for the front end of your final website.
3. *Web service code.* JavaScript code for your final web service. **Please do not include the node_modules folder in your submission. It could make it too large to upload!**
4. *Database dump*. Dump of the database and data using mongodump, Compass or similar tool. No marks will be awarded for a copy of the raw database files - your dump must be mostly readable in a text editor.
5. *5-minute video demonstration*. We will use the video demonstration to mark your project. **You will lose marks if you do not submit a video demonstration or if you do not show all of your code working in your video demonstration.** I strongly recommend that you watch the talk on recording video demonstrations on the course website.

Upload the zip file using the link in the Coursework 2 section of the course website.

The final submission will be marked online. It is worth 60% of the marks for Coursework 2.

5. Formative Feedback

Formative assessments do not directly contribute to the overall module mark, but they do provide an important opportunity to receive feedback on your learning. They provide an opportunity to evaluate and reflect on your understanding of what you have learnt. They also help your tutors identify what further support and guidance can be given to improve your grade.

We are happy to give you feedback about Coursework 2 in the labs and can give feedback about drafts of Coursework 2 that are sent to us more than one week prior to the deadline.

6. Extenuating Circumstances

If you have personal problems that interfere with your studies, you can apply for extra time to complete coursework without a mark penalty. You have to provide appropriate documentary evidence.

More information here: https://unihub.mdx.ac.uk/your-study/assessment-and-regulations/extenuating-circumstances.

You must let the module leader know if you have been granted an extension.

7. Late Submission

We are very unlikely to give extensions to coursework and very unlikely to accept excuses. So we strongly recommend that you hand coursework in on time. Contact the module leader before the deadline if you run into problems.

Each submission has a *two-hour window* (16:00-18:00) within which you can upload your work without a mark penalty. If you submit later than this (within five calendar days of the original deadline), then your mark will be capped at 40% of the marks available. Work submitted more than five days after the original deadline will receive zero marks.

Students registered with the Disability and Dyslexia Service, who have a Learning and Support Form (LSF), can submit within five calendar days of the original deadline without capping.

8. Plagiarism

Plagiarism is a serious academic offence. Students that submit identical projects will be reported to the university. If they are found guilty, they will have to resubmit their work, their marks could be capped or they could fail the module.

We recognize that there is often a blurry line between copying and collaboration. People work together and help each other to solve problems and apply the solutions to their own projects. We strongly encourage this kind of collaboration. But it is not acceptable for students to collaborate on a project which they submit as individual work. To penalize this, the mark for near-identical projects will be divided between the projects. So suppose a project gets a mark of 60% and near-identical versions are handed in by 3 people. Each person will get 20%, instead of 60%. This only applies to the marks for the parts of the project that are identical.

We are not going to police this and make detailed investigations. So if you allow your project to be copied, you will be as liable for plagiarism as the person who submits it as their own work. Both the original and the copy will receive zero or reduced marks. Links to the relevant University regulations and additional support resources can be found here:

- Full details on academic integrity and misconduct and the support available can be found at: https://unihub.mdx.ac.uk/study/academic-integrity.
- The Academic Integrity and Misconduct policy is available in our Public Policy Statements (under Academic Quality) at: https://www.mdx.ac.uk/about-us/policies.
- Referencing & Plagiarism: Suspected of plagiarism?: https://libguides.mdx.ac.uk/c.php?g=322119&p=2155601.
- Referencing and avoiding plagiarism: https://unihub.mdx.ac.uk/study/writing-numeracy/awl-resources/writing.

The Middlesex University Students' Union (MDXSU) Advice Service offers free and independent support in making an appeal, complaint or responding to any allegations of academic or non-academic misconduct: https://www.mdxsu.com/advice.

9. Assessment Methods

9.1 Project Proposal

We will read your project proposal and look at your database design.

9.2 Front End and Basic Web Service

We will read the documentation of your front end and basic web service. We will also look at your code and may run it to check that it works.

9.2 Final Submission

We will look at your code, read your report and view up to 5 minutes of your video demonstration. Your video demonstrations should not be significantly longer than 5 minutes**. You will lose marks if you do not submit a video**

demonstration or if you do not show all of your code working in your video demonstration. I strongly recommend that you watch the talk on recording video demonstrations on the course website.

The project will be given a mark out of 100. This will be scaled down to a mark between 0 and 50 that corresponds to 50% of the overall mark for the module.

10. Assessment Criteria

Feature	Deadline	Marks
Project proposal. A short description of the project that includes wireframes of the design or screenshots of an early prototype.	**16:00 26/1/24**	**2 marks**. Description of functionality of website. **2 marks**. Wireframes or screenshots of website. **2 marks**. Proposal quality. Is it clearly written? Do the wireframes clearly show the website design?
Mongo DB database design. List the collections and give examples of documents within each collection (for example, Users, Friends, Blogs, etc.). You can paste document examples into the progress report or include them in a zip file with the proposal. A database dump of an implementation of the design can also be included with the proposal.	**16:00 26/1/24**	**4 marks.** List of collections and examples of documents within each collection. Does the design capture all the data that is required for a social networking website?
Front-end. A single HTML page implemented in HTML, CSS and JavaScript. **No marks are available for websites with multiple HTML pages.** **No marks are available for a front end implemented using a banned third-party framework, such as React or Vue.**	**16:00 23/2/24**	**7 marks.** Attractiveness and usability. Is the front-end well designed and attractive? Has thought been given to usability?
Basic web service. A basic web service that enables the user to store and retrieve JSON formatted data using the GET and POST or PUT HTTP methods. The web service must be implemented using Node, Express and MongoDB. All the data that is sent and received must be in JSON format. **The path must start with your student ID.** For example, http://localhost:8080/M0012345/. **At least one of the stored and retrieved documents must contain your name, student ID and student email address.**	**16:00 23/2/24**	**3.5 marks.** Data sent from client to server in JSON format using HTTP POST or PUT. The path must start with your student ID. **3.5 marks.** Data sent to server is stored in MongoDB. One of the documents must contain your name, student ID and student email address. **3.5 marks**. Data sent from server to client in JSON format in response to HTTP GET request. The path must start with your student ID. **3.5 marks.** Data sent to user is retrieved from MongoDB. One of the returned documents must contain your name, student ID and student email address.
HTML and CSS code quality. Your code should be well commented, tidy and easy to read. Files	**16:00 23/2/24**	**1.5 marks.** HTML code quality

should be sensibly organized into folders. Marks will be deducted for unused files and commented out code.		**1.5 marks.** CSS code quality
Front end and web service documentation. Screenshots of front end of website; screenshots of Postman showing data sent and retrieved to/from web service; screenshots of MongoDB showing stored data. Must be in Word or PDF format. **Marks may be lost if documentation is incomplete.**	**16:00 23/2/24**	**2 marks.** Screenshots of front end of website **2 marks.** Screenshots of Postman showing GET and POST/PUT functionality working. The data sent and received must be shown in the screenshots. **2 marks.** Screenshots of MongoDB showing stored data.
Registration. User can register on the website. Registration data must be sent to the web service in JSON format using the HTTP POST or PUT method. An acknowledgement with a possible error message (for example, missing data) should be returned in JSON format. User data is stored in MongoDB.	**16:00 12/4/24**	**2 marks.** User enters registration data on front end. Client sends registration data to web service in JSON format using HTTP POST or PUT method. Error message returned in JSON format if registration fails. **2 marks**. User registration data stored in MongoDB.
Login. User can log in to the website. Data is sent to the web service in JSON format using the HTTP POST or PUT method. An acknowledgement with possible error message (for example, incorrect password) should be returned in JSON format. Login feedback displayed to user on front end. Login status recorded with session management.	**16:00 12/4/24**	**1 mark.** User enters data on front end. Login data sent in JSON format to web service using POST or PUT method. **2 marks**. Appropriate feedback sent back to user in JSON format – password incorrect, login successful, etc. This information is displayed to user. **2 marks**. Session management correctly used to track user's login status across multiple HTTP requests.
Contents. User can upload contents (text, images or files) to the website. Text data must be submitted in JSON format using the HTTP POST or PUT method. An acknowledgement with possible error message (for example, missing data) should be returned in JSON format. Image and file contents do not need to be uploaded in JSON format.	**16:00 12/4/24**	**1 mark.** User uploads text contents, such as blog or recipe, to server in JSON format using HTTP POST or PUT method. **1 mark.** Text data uploaded by user correctly stored in MongoDB. **1.5 marks.** User uploads image or file to server using HTTP POST or PUT method. **1.5 marks.** Image or file stored on server along with correct entry in MongoDB.
Search. User can search for other users to follow and for content that has been uploaded by other users. Search requests must be submitted in JSON format using the HTTP GET method. Results must be returned in JSON format.	**16:00 12/4/24**	**2 marks.** User can search for people to follow. **2 marks**. User can search contents of posts, recipes, etc.
Social Networking. User can choose to follow other users and selectively view their uploaded	**16:00 12/4/24**	**2 marks.** User can follow people. Follow request is submitted in JSON format to web

content in a single page. There is no need to implement friend requests that have to be accepted by the recipient.		service using HTTP POST or PUT method. **2 marks.** User can exclusively view posts, pictures, etc. that have been uploaded by the users that they are following. Content is sent from server to client in JSON format in response to HTTP GET request.
Advanced website functionality. These marks are for websites whose functionality exceeds the minimum requirements. Marks will be awarded for better implementations and more complex projects. Examples include, but are not limited to, input checking, error handing, LinkedIn style friend requests, feed sorting and filtering.	**16:00 12/4/24**	**10 marks**. Marks awarded for functionality that substantially exceeds the minimum requirements that are specified in this document.
JavaScript code quality. Your code should be well commented, tidy and easy to read. Files should be sensibly organized into folders. Marks will be deducted for unused files and commented out code.	**16:00 12/4/24**	**3 marks.** JavaScript code quality.
Testing. The results of the tests must be documented in your project report. **Your code must pass the tests. Marks will be halved for failed tests. Undocumented tests will be assumed to fail.**	**16:00 12/4/24**	**5 marks.** HTML and CSS validation. Full marks will be awarded for complete coverage of the website. **5 marks.** JavaScript tests (1 mark per test).
Security, Privacy Legal Issues. Discussion of the security, privacy and legal issues that affect the website and how they could be fixed.	**16:00 12/4/24**	**5 marks**. Discussion of the security, privacy and legal issues that affect the website. List the steps that have been taken to address these issues and suggest how unresolved issues could be addressed in the future. Should be at least 500 words.
Database dump. Dump of the database and data using mongodump, Compass, or similar tool. No marks will be awarded for a copy of the raw database files. **Your dump must be mostly readable in a text editor**.	**16:00 12/4/24**	**4 marks.** Complete dump of database.
Project report. Brief description of the project. Documentation of web service and screenshots of the final front end of the website. **Do not include screenshots of code, the command line, Dreamweaver, VSCode, etc.**	**16:00 12/4/24**	**2 marks**. Screenshots of all the website's front end functionality. **2 marks.** Screenshots of Postman showing all the functionality of your final web service. The data sent and received must be visible in the screenshots. **2 marks**. Content of report. Does it clearly describe the project?