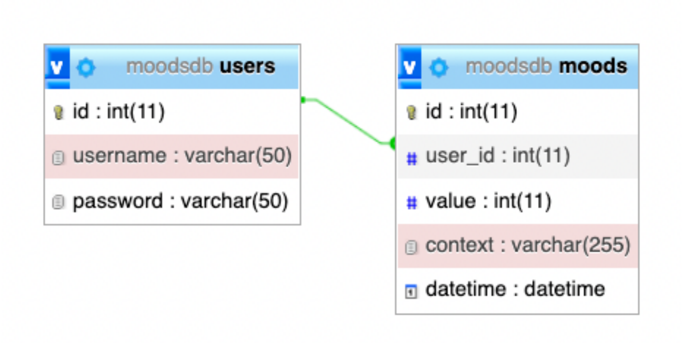
Web Development Assessment Overview

The purpose of this supplementary assignment is to assess your skills in designing, developing, and testing a bespoke website for a given specification using client-side and server-side development technologies. The website must be implemented using your choice of the core technologies covered in the module (e.g., HTML, CSS, JS, PHP, Node JS, Express JS).

A set of elementary MySQL database tables containing no data has been provided (*moodsdb.sql*) which may be used/modified for the assessment, though you are free to create/use your own MySQL database:



moodsdb **users**
- id : int(11)
- username : varchar(50)
- password : varchar(50)

moodsdb **moods**
- id : int(11)
- user_id : int(11)
- value : int(11)
- context : varchar(255)
- datetime : datetime

Please note: this assignment contributes 100% of the overall assessment for the module (any re-sit submissions will be capped at 50% whereas for any deferral submissions the mark will be uncapped).

Generic Website Specification:

Given the rising public awareness of the potential benefits of emotional wellbeing, there has been growth in the use of approaches to support activities aimed at recording and measuring emotional wellbeing, notably through activities such as journaling, meditation and improving mindfulness.

Consequently, the bespoke website is a mood tracking website that tracks the daily mood(s) of the user and records this for subsequent review in an effort to facilitate insight into the user's moods over time. The website will record the user's mood at arbitrary times throughout the day, allowing them to build a mood profile over time. Additionally, the website will enable the user to optionally provide some context for each mood entry by allowing the user to input a possible trigger for the recorded mood. Subsequently, the website will enable the user to view the changes in their moods over time and identify any related context, such as the triggers that lead to positive or negative moods.

Typically, the user will access the website and start a *mood logging session*, after which they will be prompted to enter their current mood. The user will then be able to optionally enter other contextual information that contributed to their current mood. Both the current mood and contextual information will then be stored by the website. Such a mood logging session may be carried out one or more times during a day. The user can view a *mood list* of their previously recorded moods, which permits the user to add or edit contextual information, or to delete an individual mood entry. At any stage, the user can also view a *mood summary* of their recorded moods over time, providing insight into their changing moods.

Consequently, the **functional requirements** for the website are as follows:

When accessing the website, the user should be able to:

I. **User Registration**: A new user should be able to register an account on the website.

II. **User Login**: A registered user should be able to login with an existing account on the website to permit access to the website functionality.

Once logged into the website, the user should be presented with the options:

III. **Mood Logging**: When this option is selected, the user should be given the opportunity to begin a mood logging session. The information recorded during the event should include the *current date*, *current time*, *current mood* and *optional mood trigger*.

IV. **Mood List**: When this option is selected, the user should be able to view a list of previously recorded mood logs, including any corresponding contextual information. Subsequent options for the mood list are as follows:

    a. **Edit Mood**: The user should be able to select a previously recorded mood log from the mood list and edit or add contextual information to the mood. The value specified for the mood entry should not be changed.

    b. **Delete Mood**: The user should be able to delete a previously recorded mood log from the mood list.

V. **Mood Summary:** When this option is selected, the user should be shown a summary visualization of their recorded moods over time.

In addition, when logged into the website, the user should be able to:

VI. **Delete Account**: A registered user should be able to delete their account, removing all corresponding credentials and mood logging entries stored by the website.

VII. **Logout**: A logged in user should be able to logout of the website.

In developing the website, additional **system requirements** are as follows:

VIII. The implementation of the website should attempt to incorporate a REST API that allows the front-end to push and pull data to/from the back-end MySQL database using the appropriate REST API verbs.

IX. The implementation of the website should provide appropriate validation of user input, along with appropriate authentication and address potential security vulnerabilities.

There are many ways in which the requirements can be approached and many design decisions to make. For example, beyond the overarching UI and UX for the website, the set of possible moods and mood triggers utilised by the website need be identified from the outset. Likewise, there are many ways to potentially display a mood summary, such as using statistics and/or visualizations, with the choice of statistics and/or visualizations to be determined by the developer. To fulfil this assignment brief, you are required to complete the following tasks:

Task 1: Design, develop and test a ***website*** based on the given specification using both client-side and server-side technologies covered in the module. Note: you are permitted to use client-side libraries and frameworks, e.g., Bootstrap, Bulma, etc., however, you are not permitted to use pre-built templates or server-side frameworks beyond Node JS and Express JS, e.g., Laravel, CakePHP, Symfony, CodeIgniter, etc.

Task 2: Produce a ***report***, no more than 7 A4 pages in length, which details the design decisions made regarding the implementation of the website. Guidelines outlining the content required in the report can be found in the *Report Guidelines* section later in this assignment specification.

Task 3: Produce a ***demonstration video***, lasting no more than 15 minutes, which provides a narrated walkthrough of your website, briefly showing all successfully implemented functionality, and brief narrated walkthrough of the corresponding source code and database contents. Guidelines outlining the content required in the demonstration video can be found in the *Demonstration Video Guidelines* section later in this assignment specification.

Subsequently, the corresponding deliverables for this assignment (as detailed in the *Assignment Deliverables* section later in this assignment specification) must be submitted by the following deadline:

Thursday 23rd March 2023 11.59pm

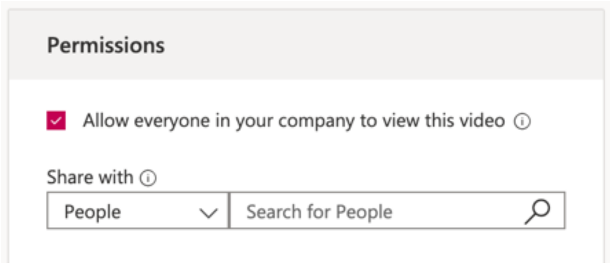
With the exception of students that have been granted an extension by the Exceptional Circumstances panel, or where flexibility with deadlines has been agreed under an Individual Student Support Agreement (ISSA), a penalty for late submissions will be applied as follows:

“Coursework submitted after the deadline will be penalised at the rate of 5% of the total marks available for each calendar day late up to a maximum of five calendar days, after which a mark of zero shall be awarded, i.e., up to one calendar day is 100% - 5%; up to two calendar days is 100% -10%; up to three calendar days is 100% - 15%, etc.”

Note: if you have a suitable ISSA in place, please contact the module owner to request flexibility with the submission deadline if necessary.

Assignment Deliverables

(1) Demonstration Video Link. Upload your demonstration video to a video sharing platform such as YouTube or Microsoft Stream. You must allow all internal and external markers to view the video, so ensure it is viewable by setting it as an “Unlisted” if using YouTube, or by ensuring the check box “Allow everyone in your company to view this video” is ticked if using Microsoft Stream:



Permissions

☑ Allow everyone in your company to view this video ⓘ

A link to the corresponding shared, published video must be given in the report submission. It is your responsibility to ensure can be clearly viewed with sound. In accordance with university assessment guidelines, demonstration videos that go beyond 15 minutes will be penalized as follows:

15 minutes +10%: no penalty
15 minutes + >10% - 20%: reduction in total mark by 10%
15 minutes + >20% - 30%: reduction in total mark by 20%
15 minutes + >30% - 40%: reduction in total mark by 30%
15 minutes + >40% - 50%: reduction in total mark by 40%
15 minutes + >50%: maximum total mark achievable is 50%

(2) Report. Upload your report in PDF format via the ***Report Submission*** link on Canvas (within Assignment link). The report should be named *yourstudentnumber*.pdf (i.e., 12345678.pdf). It is your responsibility to ensure the PDF can be opened and viewed. Reports submitted in formats other than PDF won’t be accepted by the submission link.

(3) Source Code. A compressed (**zip**) file containing the project folder(s) with source files and resource files used for the implementation of the website, including exported database tables, should be uploaded via the ***Source***

Submission link on Canvas (within Assignment link). The zip file should be named *yourstudentnumber*.zip (i.e., 12345678.zip). It is your responsibility to ensure the zip file can be successfully decompressed after download.

Note: Both deliverable (2) and (3) must be submitted via Canvas for this assessment, with deliverable (1) included within deliverable (2). Failure to submit a deliverable will result in a mark of 0%. Students must ensure submitted materials can be successfully opened and viewed (with sound in the case of the demonstration video). Corrupt files will be treated as a non-submission.

Report Guidelines

The following guidelines outline the expected content of the report, which complements the demonstration video, that details the rationale and justification for decisions made during the design and implementation of the website:

1. Name, Student Number and QUB email.

2. URL for publicly accessible demonstration video.

3. Brief discussion and justification for the choice of moods and mood triggers used during a *mood logging* session. In addition, a brief discussion and justification for the choice of statistics and/or visualization approach used for a *mood summary* should also be given. References to supporting sources of information should be given if applicable.

4. Brief discussion of the final set of implemented functional and system requirements, including supporting rationale for any design decisions made in relation to the requirements if not previously given.

5. Brief overview of any implemented authentication and security measures, along with brief identification and rationale for the choice of credentials used for user login and registration.

6. Overview of the REST API implemented, including a list of the API endpoints available along with any API keys used if applicable.

7. ER diagram of the final MySQL database used, along with a brief explanation of any modifications made to the provided dataset if applicable.

8. Brief discussion on the underlying structure employed for the implementation of the website, which includes a listing of the source files and folders used and gives a brief rationale for any client-side and server-side libraries or frameworks utilized. References for any libraries or frameworks used should be given.

The report should use a 12pt font size, single line spacing and be saved as PDF format.

Demonstration Video Guidelines

The following guidelines outline the expected content of the demonstration video walkthrough for the implemented website and underlying source code:

1. Clearly demonstrate and provide narration for all successfully implemented requirements by showing the related functionality in operation on the website, along with evidence of any resulting data persistence that has occurred. Include examples of any form validation, user authentication and security considerations employed in the operation of the requirements, and briefly demonstrate the responsiveness of the website using Google Chrome Developer Tools.

   As a guide, when demonstrating the functional requirements:

   i. create a new user account to demonstrate *user registration*, *user login*, *mood logging* and *mood summary* requirements prior to demonstrating the *delete account* requirement

   ii. use a pre-existing user account with pre-populated mood entries to demonstrate *user login*, *mood list* (*edit mood*, *delete mood*), *mood summary* and *logout* requirements

   You may choose how the features are demonstrated in terms of the target users; the key is to ensure that you have clearly shown all requirements that have been successfully implemented along with any corresponding changes to data stored in the database tables.

2. Briefly show and explain key parts of the underlying source code used during the implementation of the requirements, including details of any form handling, event handling, and related processing that occurs, along with the implementation of any validation, authentication, error handling and security considerations. Where applicable, show and explain key SQL queries, along with REST API request methods implemented.

3. Clearly demonstrate any REST API verbs successfully implemented, if applicable, by showing and briefly explaining examples of the corresponding request methods being performed along with their responses using Postman.

4. Provide a brief overview of the project structure.

The demonstration video must include a live narrator view, that shows you in the bottom corner of the screen as you narrate the walkthrough. Ensure that the narrator view does not occlude any key information currently being discussed (e.g., UI controls, source code, database content, etc.)

Marking Criteria

Marking will be conducted based on the information provided in the report and demonstration video.

An indicative marking rubric is provided at the end of this assignment specification, with the following component weightings:

Assignment Component	Weighting
User Interface Design	15%
Requirements Implementation	40%
REST API Implementation	30%
Project Structure	10%
Report & Demonstration Video	5%

User Interface Design [15%]

Poor [0% - 49%]	Good [50% - 59%]	Very Good [60% - 69%]	Excellent [70% - 79%]	Outstanding [80% - 100%]
Basic UI demonstrated, lacking consistency or responsiveness. UI lacks sufficient consideration and justification of relevant design choices.	Functional UI demonstrated, which is mostly consistent and responsive. UI incorporates elementary consideration and some limited justification of relevant design choices.	Functional UI demonstrated, which is fully consistent and responsive. UI incorporates good consideration and justification of design choices.	Functional, intuitive UI demonstrated, which is fully consistent and responsive. UI incorporates excellent consideration and justification of design choices.	Functional, intuitive, and creative UI demonstrated, which is full consistent, and responsive. UI incorporates outstanding consideration and justification of design choices.

Requirements Implementation [40%]

Poor [0% - 49%]	Good [50% - 59%]	Very Good [60% - 69%]	Excellent [70% - 79%]	Outstanding [80% - 100%]
Limited set of functional requirements and corresponding data persistence successfully demonstrated (e.g., I, II, III, VII). Basic attempt to implement a solution to validation, authentication, and security concerns.	Primary functional requirements and corresponding data persistence successfully demonstrated (e.g., I, II, III, IV, VII). Good attempt to implement a solution to validation, authentication, and security concerns.	Majority of functional requirements and corresponding data persistence successfully demonstrated (e.g., I, II, III, IV, V, VI, VII), incorporating simplistic mood summary. Very good attempt to implement a solution to validation, authentication, and security concerns.	All functional requirements and corresponding data persistence successfully demonstrated (e.g., I, II, III, IV, IV(a), IV(b), V, VI, VII), incorporating complex mood summary. Excellent attempt to implement a solution to validation, authentication, and security concerns.	All functional requirements and corresponding data persistence successfully demonstrated (e.g., I, II, III, IV, IV(a), IV(b), V, VI, VII), incorporating complex, creative mood summary. Outstanding attempt to implement a solution to validation, authentication, and security concerns.

REST API Implementation [30%]

Poor [0% - 49%]	Good [50% - 59%]	Very Good [60% - 69%]	Excellent [70% - 79%]	Outstanding [80% - 100%]
REST API request methods not implemented or malfunctioning. Limited or no demonstration of request methods using Postman. Website implementation primarily utilises SQL queries to provide direct access to MySQL database.	REST API GET and POST request methods working and successfully demonstrated using Postman. Website implementation utilises SQL queries to provide some direct access to MySQL database, along with successful attempt to incorporate REST API for limited number of database interactions.	REST API GET and POST request methods working and successfully demonstrated using Postman. Website implementation successfully incorporates REST API for all interactions with MySQL database.	REST API GET and POST request methods working and successfully demonstrated using Postman, along with additional and appropriate PUT/PATCH or DELETE request method. Website implementation successfully incorporates REST API for all interactions with MySQL database.	REST API GET, POST, appropriate PUT/PATCH, and DELETE request methods working and successfully demonstrated using Postman. Website implementation successfully incorporates REST API for all interactions with MySQL database.

Project Structure [10%]

Poor [0% - 49%]	Good [50% - 59%]	Very Good [60% - 69%]	Excellent [70% - 79%]	Outstanding [80% - 100%]
Project has monolithic structure. Major issues with underlying source code result in many warnings/errors appearing on website. Lack of rationale given for libraries/frameworks utilized.	Project has elementary structure. Minor issues with underlying source code result in occasional warnings/errors appearing on website. Limited rationale given for libraries/frameworks utilized.	Project has elementary, modular structure. No issues with underlying source code, with no warnings/errors appearing on website. Good rationale given for libraries/frameworks utilized.	Project has fully modular structure. No issues with underlying source code, with no warnings/errors appearing on website. Excellent rationale given for libraries/frameworks utilized.	Project has efficient, fully modular structure. No issues with underlying source code, with no warnings/errors appearing on website. Outstanding rationale given for libraries/frameworks utilized.

Report & Demonstration Video [5%]

Poor [0% - 49%]	Good [50% - 59%]	Very Good [60% - 69%]	Excellent [70% - 79%]	Outstanding [80% - 100%]
Report lacks adherence to required content and/or format. Demonstration video has poor quality audio/visual content. Content of video lacking adherence to guidelines and/or insufficient detail and/or visual presence.	Report mostly adheres to required content and format. Demonstration video only partially covers relevant points given in guidelines but suffering from some obvious gaps/issues with structure/content.	Report fully adheres to required content and format. Demonstration video mostly covers relevant points given in guidelines but suffering from some obvious gaps/issues with structure/content.	Report fully adheres to required content and format. Demonstration video covers all relevant points given in guidelines but suffering from some issues with structure/content.	Report fully adheres to required content and format. Demonstration video is well-structured, covering all relevant points given in guidelines in detail.