

## Assignment Overview

This assignment aims to assess your understanding and knowledge in the design and implementation of a dynamic web app for a given specification. The web application must be implemented utilising your choice of the core client-side and server-side technologies covered in the module (e.g., HTML, CSS, JavaScript, Node JS, Express JS).

**Please note:** this is an individual assignment that 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).

### Emotion Tracking Web App:

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 and meditation to improve mindfulness.

Consequently, to provide a platform to support users' awareness of their emotional wellbeing, the bespoke web app to be developed focuses on *emotion tracking* that permits a user to record snapshots of their emotional state at arbitrary times throughout the day for subsequent review in an effort to facilitate insight into the user's emotions over time. Accordingly, a single snapshot of a user's emotional state will contain values indicating the level of emotion currently being experienced for each of the seven universal emotions: **enjoyment, sadness, anger, contempt, disgust, fear, surprise\***. In addition, the snapshot will also contain a **timestamp** indicating the date and time of the snapshot, along with an optional set of possible **triggers** that may provide context for the emotional state recorded in the snapshot. For example, an emotional snapshot may have been triggered by a user's *social interaction, physical activity, family, work, sleep, weather*, etc. Subsequently, the web app will enable the user to view changes in their levels of universal emotions over time and identify any related contextual triggers.

In terms of the overall user flow, a user will access the web app and start an 'emotion logging session', after which they will be prompted to enter values for the current strength of each of the seven universal emotions. Once the level of each emotion has been specified, the user will then be able to optionally provide other contextual information relating to potential triggers that may have contributed to the current emotional state, then save the current snapshot. Such an emotion logging session may be carried out by the user one or more times during a day. The web app will also permit a user to login to their account and view a list of any previously recorded snapshots, which permits the user to add or modify the contextual triggers for an individual snapshot, or to delete an individual snapshot. At any stage when logged in, the user can also view a

---

\* <https://www.paulekman.com/universal-emotions/>

summary visualization of their recorded emotional states over time, providing insight into their overall emotional status and changing emotions.

### **Web App Requirements:**

A modern, responsive data-driven web app should be designed, developed and tested that provides the following functionality:

- I. The web app should facilitate multiple users such that individual users should be able to register, login and logout. Data specific to an individual user should only be accessible to the corresponding user.
- II. The web app should provide appropriate validation of any user input where necessary, along with appropriate user authentication and address potential security vulnerabilities.
- III. When logged in, a user should be able to record a snapshot of their current emotional state. The information recorded by the snapshot should include values for the user's current levels of *enjoyment*, *sadness*, *anger*, *contempt*, *disgust*, *fear*, and *surprise*, along with the current *date* and *time*. The snapshot should also include an optional set of *contextual triggers*. The choice of how the emotion levels and contextual triggers are recorded is to be determined by the developer.
- IV. When logged in, a user should be able to view a list of all previously recorded snapshots.
- V. The user should be able to select an individual snapshot from the list and change (e.g., add or modify) contextual triggers associated with recorded levels of emotion, or delete the snapshot.
- VI. When logged in, a user should be able to view a summary visualization of previously recorded emotional states over time, along with related, relevant statistics or information. The choice of summary visualization and related statistics or information is to be determined by the developer, though should aim to help users gain an insight into their observed, changing emotions.

There are many ways in which the above functionality can be approached with corresponding design decisions to make. For example, beyond the overarching user experience of the web app, how emotion levels and contextual triggers are presented and recorded are areas where enhancements can benefit the user experience. Likewise, there are many ways to potentially display a summary visualization and provide supporting statistics/information, with related choices to be determined by the developer.

## Tasks & Deliverables:

To fulfil this assignment specification, you are required to complete the following tasks:

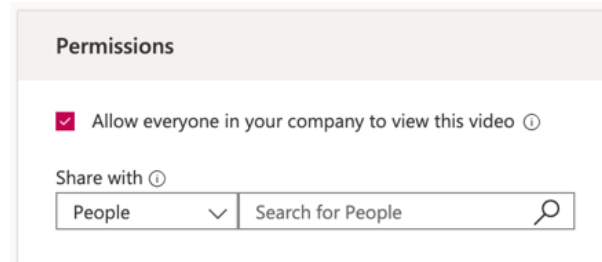
**Task 1:** Design, develop and test an interactive Node-based **Web App** to achieve the given functionality using relevant client-side (e.g., HTML, CSS, JavaScript) and server-side (e.g., Node JS, Express JS MySQL) web development technologies covered in the module. You are permitted to utilise client-side CSS libraries and frameworks, e.g., Bootstrap, Bulma, Tailwind, etc., client-side JavaScript libraries, e.g., jQuery, Chart JS, etc., or other external JavaScript Node modules installed using NPM. However, you are not permitted to use JavaScript front-end frameworks, e.g. React JS, Vue, Angular or a NoSQL database.

**Task 2:** Produce a **Presentation**, maximum 10 slides (exclusive of the title slide), which should at minimum include the following slides: *Title Slide* (project name, student name, student number), *Project Details* (URL of walkthrough video, URL of GitHub project repo (if used), test user credentials), *Database Design* (ER diagram), *Project Structure* (diagram of project files & folders), *System Implementation* (diagram and/or details covering underlying architecture of implemented web app), *Development Approach* (details about approach to development and sources used; references to libraries and any external code snippets used should be given – these do not need to include content from the CSC7084 module or media assets used).

**Task 3:** Produce a **Walkthrough Video**, maximum 20 minutes, which provides an overview of the project, followed by a narrated walkthrough of your web app's functionality and explanation of the corresponding source code. Guidelines outlining the content required in the walkthrough video can be found in the *Walkthrough Video Guidelines* section given later in this assignment specification.

Subsequently, there are three corresponding deliverables required for the submission of the assignment:

**Deliverable 1: Walkthrough Video Link.** Upload your walkthrough 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:



A link to the corresponding shared, published video must be clearly given in the Presentation component of the assignment submission.

**Deliverable 2: Exported Presentation File.** Upload a **PDF** copy of your slide set via the **Presentation Submission** link on Canvas (within the Assignment link). The presentation file should be printed/exported to PDF as one slide per page and named *yourstudentnumber.pdf* (i.e., 12345678.pdf). It is your responsibility to ensure the PDF can be opened and clearly viewed. Presentations submitted in formats other than PDF won't be accepted by the submission link. Note: the PDF of the slide set should only contain the slide content not any accompanying notes.

**Deliverable 3: Archived Project Files.** A compressed (**zip**) file containing the project folder(s) with all source code files and resource files used for the implementation of the web app, including a folder containing the exported database tables, and readme file detailing the URL for the local hosted web app, along with any build instructions, and instructions for running the web app should be uploaded via the **Source Submission** link on Canvas (within the Assignment link). The zip file should be named *yourstudentnumber.zip* (i.e., 12345678.zip). Note: the *node\_modules* folder should not be included within the project folder.

Both Deliverable 2 and Deliverable 3 must be submitted via Canvas for this assessment, with Deliverable 1 clearly 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, decompressed and viewed, where appropriate, with clear video and audible sound in the case of the walkthrough video. Corrupt files will be treated as a non-submission. Deliverables must be submitted by the following deadline:

**Thursday 14<sup>th</sup> March 2024 11.59pm**

Except for students that have been granted an extension by the Exceptional Circumstances Panel, either due to circumstances or 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."*

## Walkthrough Video Guidelines:

The following guidelines outline the expected content of the walkthrough video:

- Initially, the walkthrough video should provide an overview of the project using the Presentation slide set (Task 2) with accompanying narration for each slide. Ensure all slides are clearly introduced and displayed.

Note: you do not need to explicitly narrate URLs or references, however, these should be clearly visible in the respective slides and briefly introduced in the video.

- Clearly demonstrate and provide narration for all successfully implemented functionality by briefly showing the functionality in operation on the web app, along with evidence of any resulting data persistence that has occurred (phpMyAdmin can be used to evidence data persistence). Include examples showing any validation, user authentication and security considerations employed in the operation of the functionality, and briefly demonstrate the responsiveness of the web app using the appropriate Google Chrome Developer Tools.

Note: you may choose how the web app functionality is demonstrated; the key is to ensure that you have clearly shown all functionality that has been successfully implemented, along with any corresponding changes to data stored in the database tables. You should aim to provide a demonstration of functionality related to a new user, along with a demonstration of functionality related to an existing user using pre-populated data.

- Clearly show and provide an explanation of the underlying implementation of the web app, including details of any form handling, event handling, routing, middleware, and related processing that occurs, along with the implementation of any validation, authentication, error handling and security considerations. Show and explain key SQL queries, along with any REST API implementation and use (if applicable).

Note: prioritize complex code explanations if running short on time. Postman should be used to demonstrate the successful implementation and testing of any API endpoints used, showing the API requests running in Postman for each endpoint along with the responses received.

- Provide a brief overview of the development approach used, relating to any sources that have been used along with a brief overview of the project structure. Evidence of the use of GitHub should also be briefly given (if applicable).

Note: this aspect of the walkthrough video may be given as part of the initial narration of the Presentation slides.

The walkthrough video must include a live narrator view, that shows you in the corner of the screen as you perform initial narration of the Presentation slides. Please ensure the narrator view does not occlude any key information currently being discussed.

**Marking will be conducted based on the information provided in the walkthrough video; video content given beyond the 20-minute maximum limit will not be used in the marking of the assignment.**

### Marking Criteria & Rubric:

An indicative marking rubric based on the University's [Postgraduate Taught Conceptual Equivalents Scale](#) is provided at the end of this assignment specification, with the following component weightings:

Assignment Component	Weighting
Functionality & User Experience	40%
System Implementation & Development Approach	45%
User Interface	10%
Reporting	5%

#### Functionality & User Experience [40%]

Typically, a user will access the web app as either a guest user or a registered user, with minimal functionality offered to guest users in contrast to a registered user who will have access to the full range of functionality. The overall user experience will be predicated on the correct operation of any implemented functionality in conjunction with developer choices made around the design and implementation of the functionality. *This component of the assignment focuses on what functionality you implemented for the web app, rather than on how you implemented the functionality.*

#### System Implementation & Development Approach [45%]

During the design and development of the web app, there are several areas that show your understanding and skill with the various web technologies. These include aspects such as the underlying system architecture (e.g., the structure and integration of components of the web app), code quality (e.g., robustness, modularity, etc.), consideration of security concerns, and overall approach to the development of the web app (e.g., sources of information used, iterations of development conducted). *This component of the assignment focuses on how you implemented the web app, rather than what you implemented.*

**User Interface [10%]**

The UI of the web app should look and feel as professional in its design and implementation as possible. At minimum, a modern & responsive UI should be developed. Subsequently, the use of a CSS Framework, e.g., Bootstrap, Bulma, Material UI, Tailwind, etc., is encouraged. *This component of the assignment focuses on the overall look and feel of the web app.*

**Reporting [5%]**

Evidencing the design and development of the web app needs to be presented through a combination of the short presentation along with the walkthrough of both the functionality and its underlying implementation. *This component of the assignment focuses on the suitability of the submitted walkthrough video and slide set.*

### Functionality & User Experience [40%]

Poor [0% - 49%]	Good [50% - 59%]	Very Good [60% - 69%]	Excellent [70% - 79%]	Outstanding [80% - 100%]
Limited functionality successfully demonstrated. Limited or missing demonstration of corresponding data persistence. UX needs improved in many areas. Lack of relevant consideration for user experience.	Most website functionality and corresponding data persistence successfully demonstrated, though with some flaws, issues, or bugs. UX is generally good though has room for improvement. Limited consideration provided for user experience.	All website functionality and corresponding data persistence successfully demonstrated, with only very minor flaws, issues or bugs. UX is generally very good throughout, with good consideration provided for user experience.	All website functionality and corresponding data persistence successfully demonstrated, with no flaws, issues or bugs. UX is excellent and intuitive throughout, with excellent consideration provided for user experience.	All website functionality and corresponding data persistence successfully demonstrated, with no flaws, issues or bugs. UX is excellent and intuitive throughout, incorporating creative features and outstanding consideration provided for user experience.

### System Implementation & Development Approach [45%]

Poor [0% - 49%]	Good [50% - 59%]	Very Good [60% - 69%]	Excellent [70% - 79%]	Outstanding [80% - 100%]
Limited understanding of implementation shown. Project has monolithic structure with poor consideration of modularity. Underlying source code results in issues/bugs/flaws occurring during operation of web app. Basic attempt to implement a solution to validation, authentication, and security concerns.	Appropriate understanding of implementation shown. Project has elementary structure with basic consideration of modularity. No issues/bugs/flaws occurring during operation of web app. Acceptable consideration and implementation of validation, authentication, and security concerns.	Very good understanding of implementation shown. Project structure has good consideration of modularity, though could be further improved. No issues/bugs/flaws occurring during operation of web app. Very good consideration and implementation of validation, authentication, and security concerns.	Excellent understanding of implementation shown. Project has fully modular structure employing some use of architectural patterns. No issues/bugs/flaws occurring during operation of web app. Excellent consideration and implementation of validation, authentication, and security concerns.	Outstanding understanding of implementation shown. Project has fully modular structure employing full use of architectural patterns and related software principles. No issues/bugs/flaws occurring during operation of web app. Outstanding consideration and implementation of validation, authentication, and security concerns.



### User Interface [10%]

Poor [0% - 49%]	Good [50% - 59%]	Very Good [60% - 69%]	Excellent [70% - 79%]	Outstanding [80% - 100%]
Basic UI demonstrated, lacking consistency or responsiveness. Requires major improvements in some areas.	Functional UI demonstrated, which is mostly consistent and responsive. Requires minor improvements in some areas.	Functional UI demonstrated, which is fully consistent and responsive. Requires very minor or no improvements.	Functional UI demonstrated, which is fully consistent, responsive, and intuitive.	Functional, creative UI demonstrated, which is fully consistent, responsive, and intuitive.

### Reporting [5%]

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