Modifications for 2025 from KP marking this project in May 2024

1. Remove mongodb marking and make this instead explicitly for schema changes from last semester
2. Give 5% from discretionary mark for added functionality, not including schema changes. This will have to be shown by the student (maybe a before and after comparison?) making it easier for me to mark
3. Marks for serverless functions would be 1 new function created in say lambda OR the student deciding to take a libra to convert their existing work to Lambda from Express
4. Unit testing is showing at least 1 unit test – have to include screenshot

Open Stack Development, Semester 2 - 2024

**Angular Project**

Subject: **Open Stack Development**

Entire Project Due: **See Moodle**

Entire project worth: **70% of Final result**

How to submit your project: **Different for each part of the project (see details below)**

Design a Web Application on a topic of your choice. This is an individual project:

1. Design and develop a web application using the Angular as a front end framework
2. The application will be responsive
3. The Angular application will consume back-end data via a RESTful API that will connect to a NoSQL database for performing CRUD operations.
4. The application can make use of serverless functions (Lambda) for any new API development
5. The application may also consume data from external APIs and integrate this data into the application
6. The application will run in a Container such as Docker on a cloud platform such as Heroku, AWS, Google Cloud, etc.
7. Unit and possibly integration testing can be demonstrated
8. Analyltics usage can be demonstrated

**Part 1: Proposal (Worth 15%)**

**What do submit:** Project proposal + wireframes

**How to submit:** Upload document to Moodle using the link reading

**Angular Project >** Upload **Proposal**

**When to submit: See Moodle**

You should receive feedback from your peers on the draft and where requested feedback from your lecturer

What to include in your proposal:

1. **Existing work overview**

* Include screenshots of Angular app built in previous semester.
* Include schema around which CRUD API was developed.
* Outline/List things that are currently broken with this implementation.

**1. Project description**

Write a short description of your project idea (at least a couple of sentences on each):

* What is the idea behind the web app?
* Who are the potential users of your web app?
* What are the main features of your web app?
* How will this version of the app differ and how will it improve on what you have already delivered in previous semester (focus on features, UI, schema redevelopment etc – not on serverless, docker etc)

**2. Design**

1. Produce a set of requirements
   1. Use MoSCoW to prioritise
   2. Detail the requirements with some brief User Stories
2. Create some wireframes of your reimagined functionality

* The wireframes should align with your requirements
* Do NOT create wireframes of what existing app looks like now. Consider what it will look line with your proposed work.

1. Clarify what will be the MVP functionality that you provide for this version that you will upload in early April.

**Part 2: Implementation, Final Report, Screencast (Worth 85%)**

**What do submit:** Angular Project in zip format, final report, screencast

**How to submit:** Upload a zip of the project folder to Moodle using the link reading

**Angular Project >** **Upload Project Materials**

**When to submit: See Moodle**

* **Proposal (15 marks)**
* **Application (20 marks)**

1. Make use of a CSS framework such as Bootstrap or Material to style your application and ensure responsiveness (5%)
2. The application may consume data from external APIs and integrate this data into the application (worth 5%)
3. Can the user login as an ordinary user or administrator, show how routes might be hidden, displayed, modified or disabled (5%)
4. Demonstrate basic administration functionality. Examples might include can the admin delete/suspend a user account? Can I see how many users I have? Can I modify the price of an item for sale? (5%)

* **Front end Deployment (5 marks)**
* Your front end is deployed on a public IP.
* **Database in the cloud (5 marks)**
  + You are using MongoDB Atlas (your NoSQL is in the cloud)
* **Docker (5 marks)**
  + Front end containerised and deployed to something like Amazon, Heroku, etc
* **Serverless functions: (5 marks)**
* The front-end application will interact with Express backend API or use serverless functions
* CRUD operations will be performed on the database
* **State Management: (5 marks)**
* Some elements of state management can be applied to the application
* **Testing: (5%)**
* The application can use Unit and possibly Integration testing
* **Additional work or methods: (15%)** 
  + Using approaches, tools or techniques not covered in this class. Maybe they have been covered in other modules? Maybe you use Azure instead of AWS? Or have used CI/CD techniques while working in Project 300? You are not reinventing the wheel here – how can you apply work that you have completed with other lecturers in this module?
* **Screencast (10 marks)**

Create a screencast of no longer than 10 minutes presenting the project. Include an introduction and highlight key aspects of the project and show me how they were coded. You could use some of the techniques you have learned from Pr300 in terms of explaining problem, short demo, etc.

* **Final report (10 marks) – max 1500 words (wordcount should exclude appendices)**

Please use these headings in your final report.

* + **Introduction** - Describe the web application you have created, highlighting new features and functionality you have built.
  + **Background** - Schema development. explain how your schema has evolved and developed in the implantation. If it has not evolved, explain why this is so.
  + **Components**– show angular component diagram
  + **Architecture** - Include architecture diagram of actual deployment (with short description)
  + **Implementation vs Proposal** – if original proposal differs from the final implementation – explain why this is so. Maybe you did not get proposed work completed, or maybe you went down a different path. Either way – you need to explain.
  + **Reflection** – What went well, what did not go well? How could you do things better in the future (research/time management, etc). What would you have liked to do more of
  + **Appendix 1 – special instructions** : URLS, username/passwords, installation instructions if I need to use localhost
  + **Appendix 2 – Screenshots** : include some screenshots of key functionality (should have caption). No logins please.
  + **Appendix 3 – Testing** : If you have done any testing, include screenshots of results here/coverage report, etc. Use whatever format makes sense.
  + **Appendix 4 – Original Proposal** : include your original proposal.

Is there anything else you want to add to your report? Functionality you didn’t get to build, extra functionality that was not mentioned in the project proposal? If you don’t mention it and for some reason, I miss it – this would not be nice. Make sure you highlight anything relevant to me…