MCAST

ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)

(Note : This version is to be used for an assignment brief issued to students via Classter)

Course Title	**Part-Time Course**		**Lecturer Name & Surname**	**David Deguara**	
Unit Number & Title	**IT5-12-21p – Full Stack Development Bootcamp**				
Assignment Number, Title / Type	Developing an E-Commerce website				
Date Set	25th February 2022	Deadline Date	1st April 2022		
Student Name		ID Number		Class / Group	

Assessment Criteria	Maximum Mark
KU 1.1 - Use templates, events, services, forms and bindings to show basic understanding of the React syntax.	5
AA 1.2 - Use React to build frontend UI components.	5
SE 1.3 - Design the frontend layer appropriately together and display data from and to the user.	10
KU 2.1 - Use the various HTTP requests to demonstrate understanding of REST API endpoints.	5
AA 2.2 - Build a backend REST API service that can accept HTTP requests using Express and NodeJS.	10
SE 2.3 - Design and evaluate the best way of converting a business problem into multiple CRUD requests using Express and NodeJS.	10
KU 3.1 - Use MongoAtlas to show basic understanding of MongoDb.	5
AA 3.2 - Use Mongodb driver to connect to a database and perform transactions.	10
SE 3.3 - Evaluate the best way to retrieve large result sets from a database and show them to the user of the application.	10
KU 4.1 - Separate the frontend, backend, and database layers appropriately to show understanding of separation of concerns.	5
AA 4.2 - Integrate the frontend layer with the backend layer and the backend layer with the database layer.	10
SE 4.3 - Design a robust web application that makes appropriate use of a multi-tier architecture.	15
Total Mark	100

Notes to Students:

- This assignment brief has been approved and released by the Internal Verifier through Classter.
- Assessment marks and feedback by the lecturer will be available online via Classter () following release by the Internal Verifier
- Students submitting their assignment on Moodle/Unicheck will be requested to confirm online the following statements:

Student's declaration prior to handing-in of assignment

- ❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy

Student's declaration on assessment special arrangements

- ❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.
- ❖ I declare that I refused the special support offered by the Institute.

MCAST

Developing an E-Commerce website

1.1.1 Overall Description

This assignment is split into four tasks, all of which are practical tasks. For each task, you need to use React, Node.js and Mongo DB Atlas. You can opt to use additional libraries for your project, however it is highly recommended that your lecturer is made aware of these libraries prior to submission. Should you wish to program in a functional manner, you are free to do so.

1.2.1 Submission Guidelines

- Your assignment needs to be split into two folders, namely [frontend] and [backend].
- The node_modules folders should not be included in the submission.
- Before submitting, test your application and ensure that it works.
- Compress everything into a .zip file and name it in the form of [Name/Surname/Group] – example [YourName_YourSurname_FSB_Assignment.zip]
- A 15-minute interview will take place after the submission of the assignment.

1.2.3 Interviews

Your lecturer reserves the right to interview any students on their submission. You are strongly encouraged to ensure that your submission works. If there are issues with your submission, you will be asked to provide a satisfactory explanation of your testing prior to submission. A note will be taken of your explanation. You may lose marks if your explanation is not satisfactory.

1.3.1 Task 1 – Building the Front-end (KU1.1, AA1.2, SE1.3)

Brief: Use React to design and develop a front-end for an e-commerce website. **(Total of 20 marks)**
For this task, you are required to continue developing on the front-end part of the e-commerce website started during the lectures. You are to select a specific theme for your shop and allow people to login, select items from the shop and add them to cart. The front-end should include a header containing a menu bar and a small footer showing copyright, together with the following: • A home page containing a section with current offers and promotions, a section with testimonials from other shoppers and section with a contact us form. (5) • A login and register page. (5) • A page showing a minimum of ten items that a user can purchase from the website. Each item should contain an image, a brief description and the price. (5) • A page for the shopping cart and an icon on the menu bar displaying the amount of items in the cart. (5)
KU 1.1 - Use templates, events, services, forms and bindings to show basic understanding of the React syntax.
AA 1.2 - Use React to build frontend UI components.
SE 1.3 - Design the frontend layer appropriately together and display data from and to the user.

1.3.2 Task 2 – Implementing a REST API (KU2.1, AA2.2, SE2.3)

Brief: Use Express in Nodejs to develop the back-end of the e-commerce website. **(Total 25 marks)**
For this task, you are required to continue developing on the back-end part of the e-commerce website started during the lectures. The back-end API needs to implement the following using Express in Node.js: • The user's session is saved as a cookie and has an expiry of 48 hours. (5) • The user login and register data is received from the front-end. (5) • A reply is sent back to the front-end if the user's login and registration were successful. (5) • A user's shopping cart information is sent/retrieved as required. (5) • The items sold by the website need to be sent when requested. (5)
KU 2.1 - Use the various HTTP requests to demonstrate understanding of REST API endpoints.
AA 2.2 - Build a backend REST API service that can accept HTTP requests using Express and NodeJS.
SE 2.3 - Design and evaluate the best way of converting a business problem into multiple CRUD requests using Express and NodeJS.

1.3.3 Task 3 – Setting up the Database (KU3.1, AA3.2, SE3.3)

Brief: Set up a Mongo Atlas Database for the e-commerce website. **(Total 25 marks)**
The back-end part of the website needs to be able to perform the following operations: • Register a user. (5) • Retrieve login information of a user. (5) • Save and retrieve a shopping cart. (5) • Retrieve a list of items sold by the shop. (5) • Log the number of requests received by the front-end. (5)
KU 3.1 - Use MongoAtlas to show basic understanding of MongoDb.
AA 3.2 - Use Mongodb driver to connect to a database and perform transactions.
SE 3.3 - Evaluate the best way to retrieve large result sets from a database and show them to the user of the application.

1.3.4 Task 4 – Developing a working E-Commerce Website (KU4.1, AA4.2, SE 4.3)

Brief: Develop the core functionality of the e-commerce website. **(Total 30 marks)**
Consolidate your front-end and back-end into a fully functional e-commerce website by adding: • Users can register, login and logout from the system. (5) • The password needs to be hashed prior to being stored in the database. (5) • When a user clicks on an item from the shop, it is automatically added to cart. (5) • If the item is already in the cart, its quantity is added by one. (5) • Every time there is a change in the cart, it is automatically saved inside the database (5). • When a user logs in, his cart is automatically loaded. (5)
KU 4.1 - Separate the frontend, backend, and database layers appropriately to show understanding of separation of concerns.
AA 4.2 - Integrate the frontend layer with the backend layer and the backend layer with the database layer.
SE 4.3 - Design a robust web application that makes appropriate use of a multi-tier architecture.

Important: Payment gateways are not required.