F28WP Web Programming

Lab report

Lab 3

**Student full name**:

**Student HWU username**:

**Student’s GitHub URL of the Lab**:

**Demonstrated to Lab helper**:

**Mode of demonstration**: £ face-to-face £ online

**Date of demonstration**:

**Time of demonstration**:

*The report should be about 3 pages long (excluding the title page) and submitted in PDF format. All explanations must be brief and supported by short code statements, and possibly illustrated by precise screenshots.*

1. **Create web server (2 marks)**

*Explain the structure of package.json*

*Explain how to use npm to install packages.*

*What are express and nodemon modules?*

*Explain how the HTTP server is created in your app.js*

1. **Static web pages (1 mark)**

*Explain why do we need the static middleware?*

*How does it work? Show example from the code.*

1. **Using ejs (1 mark)**

*Explain briefly ejs. Explain the structure of the ejs files (header, footer, index) and their dependence.*

*Show code excerpts from app.js that allow setting ejs and sending the static HTML content from index to the browser.*

*Explain how your contacts.ejs works.*

1. **Displaying the catalogue (2 marks)**

Draw a simple diagram to explain the dependence of the following files

* app.js
* routes/apis.js
* controllers/productController.js
* services/productServices.js
* db/productDAO.js
* db/dbQuery.js

Briefly explain this structure and the role of each file (what is it responsible for?)

Illustrate with code excepts.

What is the text displayed on the browser when the “catalog” request is successful?

1. **Ejs for dynamic content (2 marks)**

*Explain how catalogue.ejs works.*

*Using code excerpts (the function calls), show the flow of execution starting the user click on Catalog (in the web page) and ending with the actual table of products showing on the web pages.*

*Explain the different steps taken to generate this HTML that displays the table of products.*

*Mainly explain how the data is sent from the controller to the ejs file.*

Explain how you implemented the request for displaying a single article. In particularly, explain how you created article.ejs.

1. **Login (1 marks)**

*Explain your implementation of the login feature. List all files required for this functionality and explain their role.*

*Explain how the parameters (login and password) are sent by the client and processed by the server.*

*Explain the password processing in the server side.*

*Show how you used ejs to send the login result to the client.*

Part 7: **Additional features** **(1marks)**

*You need to describe only the features you implemented. Implementing correctly one feature is sufficient to get 1 mark).*

* **Listing clients**: *explain how you implemented the list of clients and how yopu protected this functionality to make it accessible only to logged user with admin privilege. This requires checking that the user is logged in and has admin privilege before allowing the execution of this use case. To. Do this, you need to use the session and set the username in the session.*
* ***Display a single client****: Explain the flow of execution and the files you added to implement the feature that allows displaying the details (all attributes) of a single client giving their id.*
* **Add product to cart**: *explain your implementation of the cart functionality. The cart is a list of products and quantities. You should explain how you set up the cart into the session.*
* **Cart management**: *explain your implementation of the cart management: modifying the quantity for a given product in the cart, deleting a product from the cart, clearing the cart.*
* **Cart display**: *explain how you calculate the price including VAT of each product and how you calculate the total price of the cart (sum of the prices of each product). Show your display of the cart with the prices, VAT and total.*
* **User checkout**: show the changes you did on the database to store the cart. Explain how you implemented the insertion of the cart into the database.