UOB CAFE

An Online Cafe portal for Students

Durgadevi Rajendran, 97481
Yuzhou Yu, 37734

Inspiration/ Purpose

University Coffee and food outlets serve as a community hub, gathering place and energy booster for students. There seems to be a continuous and growing bonding between such outlets and students which at times is impacted by the times spent in queues and times waiting for the order to be served. We, with an idea to minimise the wait times for students due to queues and order creation by the outlets, have attempted to design and create students to pre-order and pay for coffees or food from wherever they are, thereby avoiding the hazzles of standing in queues for coffee or food.

Self-Evaluation:

Item	Rating	Summary of developed application
HTML	A	Bootstrap framework has been used to generate the HTML pages. UI features with an appealing aesthetic feel have been implemented. Google map has been embedded for location finding.
CSS	A	Bootstrap framework with customisations has been used to bring out some UI effects. Enriched UI features have been styled using custom functions.
JS	A	jQuery Ajax has been implemented and an additional value-added functionality for cart creation has been done.
PNG	A	Merging, cropping and changing resolutions of images have been implemented using GIMP to create a floating 3D effect for the Logo.
SVG	A	SVG has been used for creation of the Logo enriching the UI look and feel for the Logo
Server	B	Basic Node JS web application has been used for the server side development; however, this can be extended for further additions to the functionality.
Database	B	SQLite 3 Database has been used for login functionality with both Select and Insert functions.
Dynamic Pages	A	The login and Cart functionalities have been set up as dynamic functionalities.
Depth Design	B	Payment gateway has been included for PayPal. SVG and PNG images have been created to be used in the Logo adding a strong aesthetic feel.

Working

UoB Homepage provides information about the outlet and related information through self-scrolling pages. The homepage consists of the Login and Signup functionalities where sign-up can be done with an Email ID. Once logged in, customers can select their items from an UI list of drinks and food items. These items can be added or deleted from the Cart dynamically. Once the cart is finalised, customers can then proceed to payment implemented for payment via PayPal. Payment status has been checked based on redirection other websites as the full payment function could not be tested during development. Future works section provides more details on further functionalities that could be added.

Description of Features and Functions used

HTML: Bootstrap framework has been used to generate the HTML pages. Pages have been well formatted, and we have implemented an additional unique feature of showing the map of cafes in UoB through google maps, the functionality for which has been implemented in location.html. We have also tried to implement Touch Effect functionality for achieving responsive UI.

CSS: Bootstrap, a default CSS is used to decorate the web pages. We have tried some basics animation techniques such as Shadow effects in Login and Sign up page. We have also implemented a feature-changing function when the user hovers the mouse over menu items. We have not experienced any major browser compatibility issues for the application.

JS: jQuery and Bootstrap are implemented to enhance dynamic pages and to enhance user experience. Creating a cart functionality was quite challenging but interesting development experience – we have implemented a Cart functionality that gets updated on addition\ removal of menu items by the User. We have also added an animation where the menu items flies and gets added to the basket on the click of 'Add to Cart' button. Glyphicon style font has been used.

PNG: GIMP tool has been used for creation of a floating 3D image of the logo. We have implemented this using a multi-layered image with Gaussian blurring and shadow features. Alpha channel tried for representing the transparency degree. Other basic techniques of cropping and merging have also been used.

SVG: We have used Inkscape for the first time and have attempted to create a cup and saucer image to be added to the Logo. Bezier curves, transformation and gradient were used for implementing the same. We have placed this along with the logo floating in our first page.

Server: Node.js web application framework has been used to build this website. All URLs have been validated by GET / POST functions and transfer of data over HTTP. Express.js is used to manage and handle the requests and views. Username and Password authentication are handled using Passport.js. Cookie has been used to store login information. Basic compatibility test has been done for all browsers.

Database: Sqlite3 has been used for storage and retrieval of data. This has primarily been used for the Login and Signup functionalities. The structure of table is simple and clear based on generic principles (such as BCNF). Connection between Node.js and database has been established to store and retrieve login details.

Dynamic pages: As mentioned before, jQuery and Javascript used to create a dynamic experience for users. Orders can be added to the cart and can be updated dynamically without connecting to backend. Basic UI validation also been done dynamically to enrich UI experience.

Design Depth: Design depth can be witnessed by the usage of UI tools such as GIMP, Inkscape adding an enhanced user experience and can further be witnessed in the Payment gateway function where PayPal payment has been implemented.

Steps to run the website:

1. Run **npm init** and then **npm install** to install below node modules

   Express

cookie-session
body-parser
passport
paypal-rest-sdk
sqlite3
passport-local

2. Run **node index** to host the website @ http://localhost:3000/index.html

Future Works

This application can further be extended to add the following functionalities:

- Admin or Store login can be added for completion of orders
- Cloud storage for backend and database can be implemented