SINGAPORE UNIVERSITY OF SOCIAL SCIENCES

ICT239
Web Application Development

Tutor-Marked Assignment
January 2022 Semester

TUTOR-MARKED ASSIGNMENT (TMA)

This assignment is worth 24 % of the final mark for ICT239, Web Application Development.

The cut-off date for this assignment is Monday, 02 May 2022, 2355hrs.

Note to Students:

You are to include the following particulars in your submission: Course Code, Title of the TMA, SUSS PI No., Your Name, and Submission Date.

You are required to follow the instructions from Part 1 then follow by Part 2 to complete your submission of TMA assignment, refer to the link below for more information on TMA submission requirement.

https://canvas.suss.edu.sg/courses/45713/pages/tma01?module_item_id=485948

Answer all questions. (Total 100 marks)

During this period of global COVID pandemics, the world-wide hospitality sector is greatly impacted as the global transport network is grinded to a halt due to a stricter border control.

In Singapore, businesses in hospitality sector have to innovate their business model and re-package their products.

For example, hotels have come up with staycation packages that are catered for domestic customers instead of international travellers. The package is just like any hotel booking but the only difference is the booking duration is fixed.

You are asked to design a platform via a website for consumers to register themselves to view the staycation packages available, and for them to book the package if they found a particular one of the packages is what they need.

At the same time, for the business administrator, they can login the website as an admin user to batch upload staycation packages, registered users, and booking records; and to generate a chart that summarize the incoming generated by the staycation packages over a certain period of time.

There will be two parts to this TMA, the first part mainly concerns the design and development of **V**iew and **C**ontroller components of the web application, to be implemented by HTML/CSS/JS/AJAX, Jinja, and Flask programming.

The test case is the business process when a user register and login to the website to view, select and book the staycation package.

The second part mainly concerns the design and development of the **M**odel, **V**iew, and **C**ontroller components of the apps. In addition to what are required for Q1, Blueprint, Flask-Login, NoSQL, and ChartJS programming are also required.

Question 1: View and Control (VC) Design and Development (45 Marks)

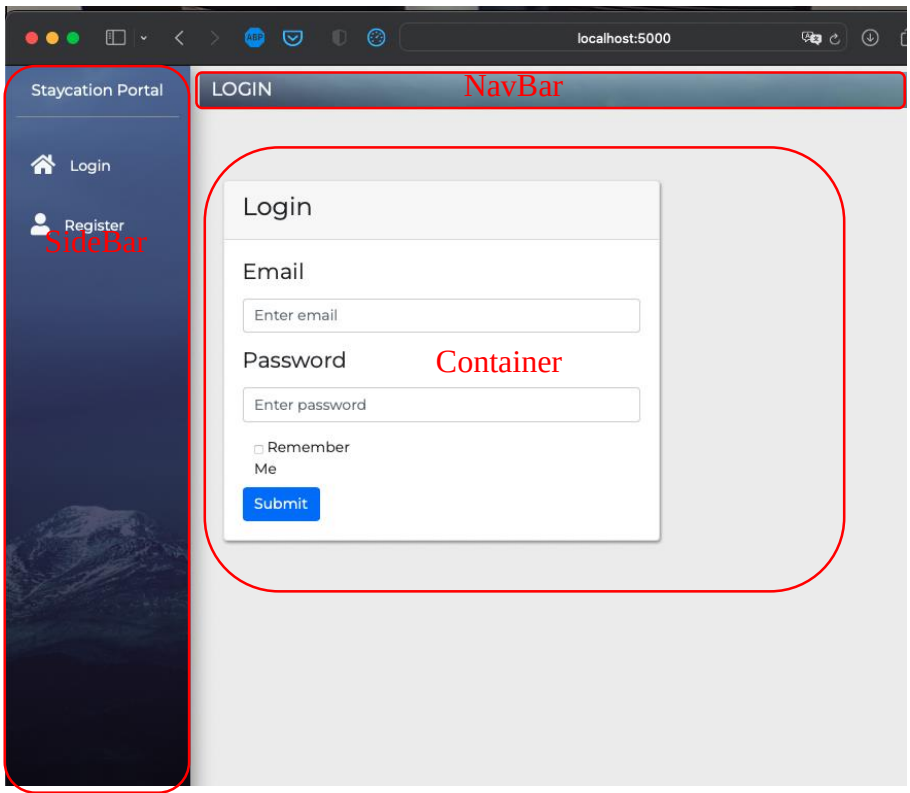

Figure Q1(a): Landing and Login Page at the url "localhost:5000/" & "localhost:5000/login"

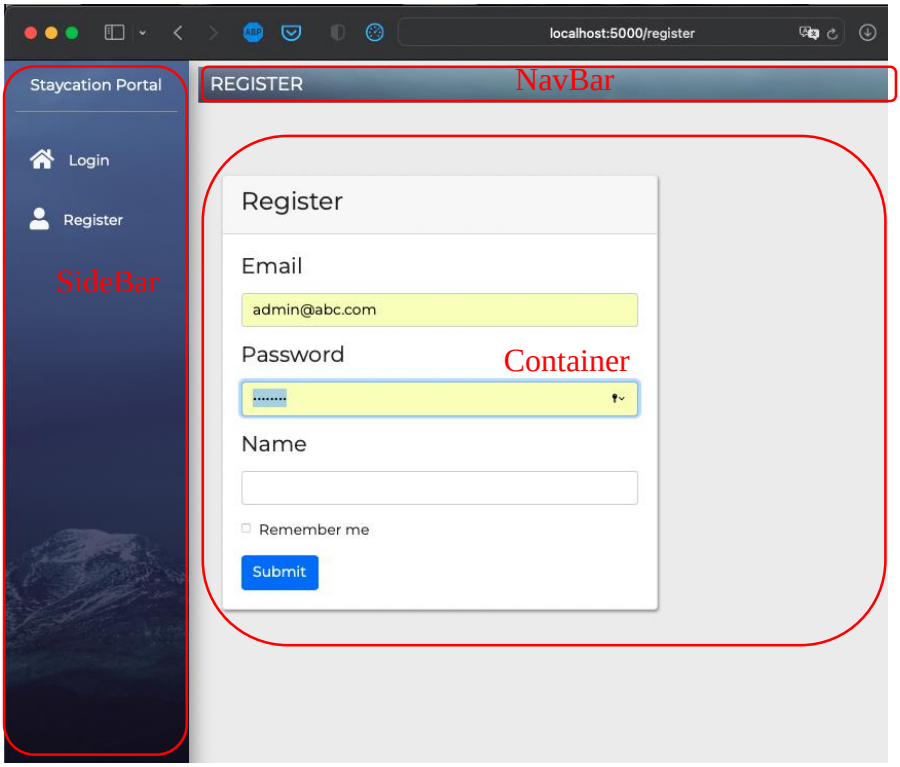

Figure Q1(b): Register Page at "localhost:5000/register"

(a) As shown in Figure Q1(a) and (b), the overall layout should follow a three component structure: NavBar, SideBar and Contrainer using the Bootstrap framework where the (1) SideBar contains linked clickable in the current view (2) the NavBar contains the name of the view and (3) the Container view is implemented in using Card of Bootstrap.

(3 marks)

(b) The Registration and Login views are as shown in Figure Q1(a) and (b).

Each should follow through with validation process on the fields of email and password, and errors message must be displayed clearly. Explain in details the logic behind checking the email and password.

(4 marks)

(c) After a user is registered and logged into the system. Construct and create the following four interfaces and its corresponding components as stated as the requirement in the part I – IV:

(i) The Sidebar will display the user name. And links to subsequent pages, However, depending on whether the user is a normal user or a admin user (here, defined by the one with email id admin@abc.com) the links displayed will be different. Normal users display only "Packages" link, while the admin user will display two additional links, "Dashboard" and "Upload".

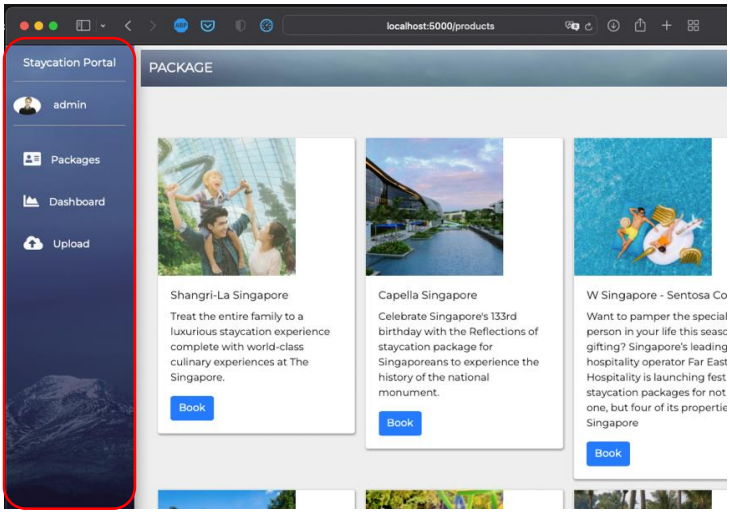

Figure Q1(c): Admin User Sidebar

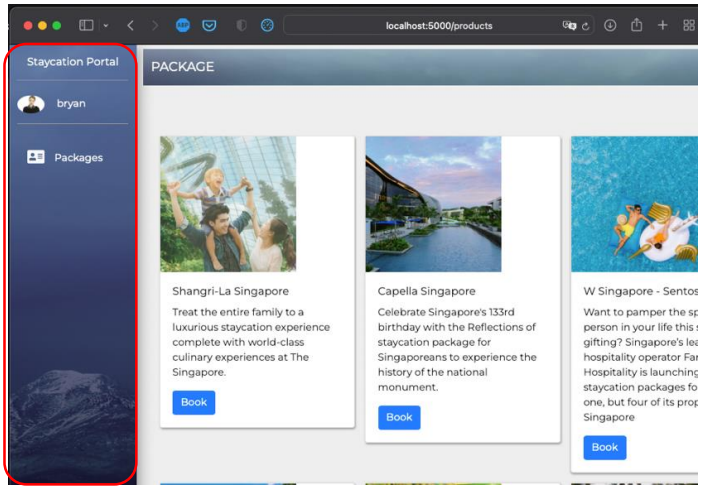

Figure Q1(d): Normal User Sidebar

(6 marks)

(ii) As shown in Q1(e), the default display after logging in for the Container is the Packages view, the packages are displayed with 3 of them in a row until all packages are displayed.

Customize the Card class in Bootstrap to display the packages accordingly. The staycation package information will be according to the content in the file "assets/js/staycation.csv".

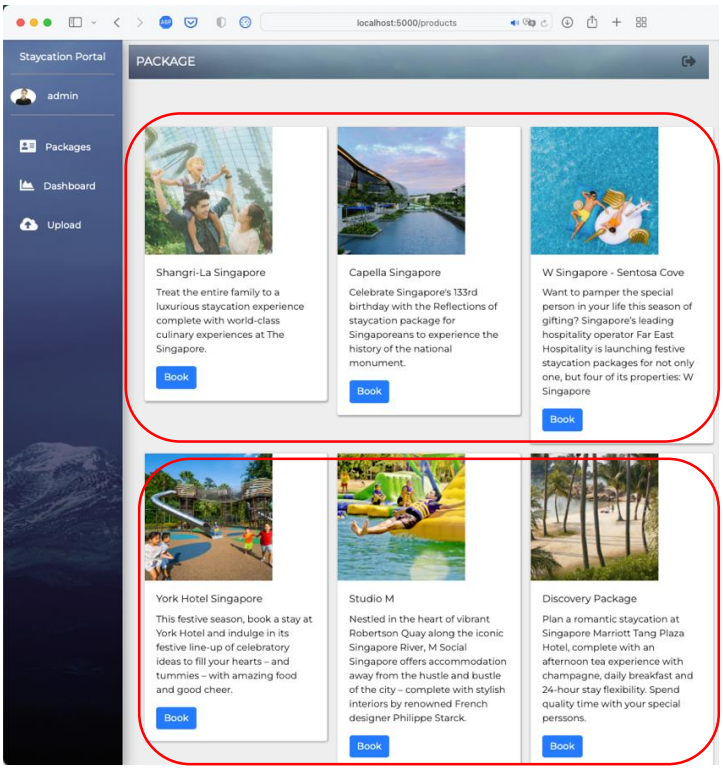

Figure Q1(e): package view with rows

(15 marks)

(iii) When the Book button is clicked, a new page view is presented as shown in Figure Q1(e) -1. The user can then specify the check-in-date for the selected staycation package as shown in the following Figure Q1(2) – 2 & 3.

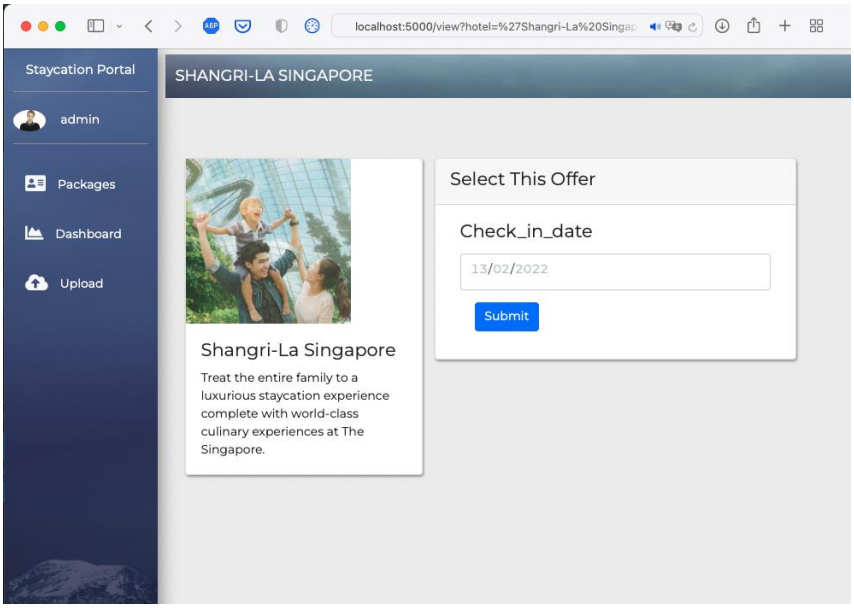

Figure Q1(e) - 1: When Book button is pressed

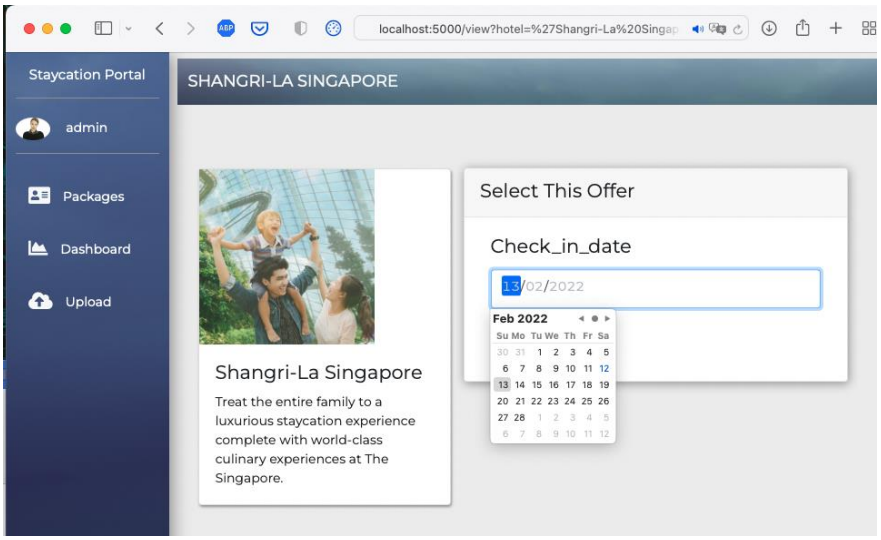

Figure Q1(e) - 2: Use the date selector to input the check in date

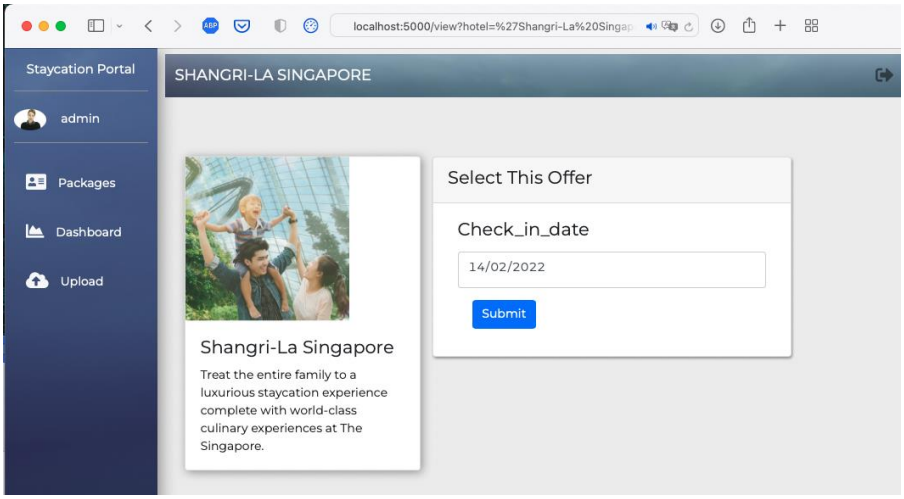

Figure Q1(e) - 3: After the Check_in_date is selected, use submit button to book the staycation package

(15 marks)

(iv) As shown in Q1(f), when the exit corner at the top-right corner is clicked, the following option will pop up for the user to logout or remain in the login state.

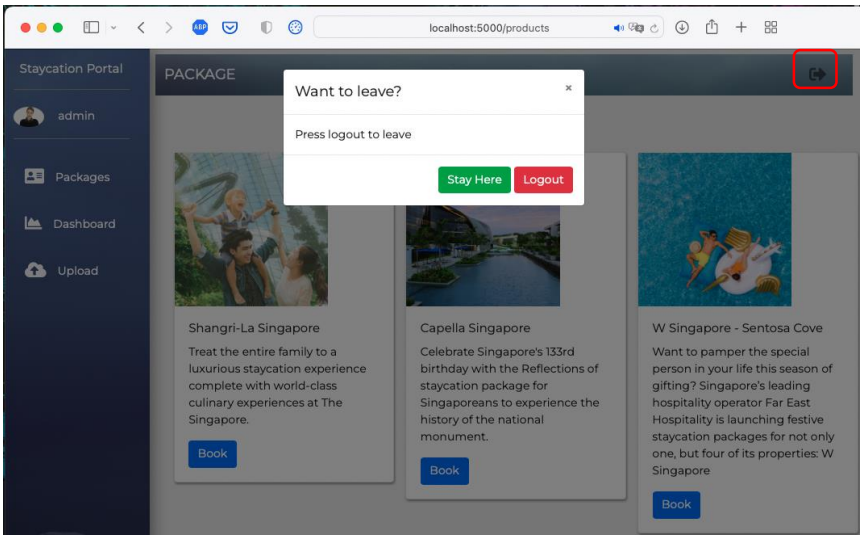

Figure Q1(f): The Logout Sidebar

(2 marks)

Question 2: **Model, View and Control (MVC) Design and Development (55 Marks)**

Question 2 concerns the design and development of the complete MVC framework.

You are required to employ the NoSQL database as the basis for the design and development of the Model component.

The main test case concerns the business processes that can be performed by an admin user (in the context of this TMA, it is hard coded as admin@abc.com) using the website.

(a) Develop the Registration and Login views and Users Model:

The frontend views of the Registration and Login should remain the same as those shown in Figures Q1(a) and Q1(b).

When the website is first started, there are no registered users and no staycations to choose from by the users. Register the system with the first admin user with the following information, and login with this admin user account.

Email	Password	name
admin@abc.com	12345	Admin

Table Q2 (a): The Schema and Content of an Admin User

Apply the Flask-login library components to control the views as shown in Figures Q1(c) and Q1(d), and NoSQL database to store the User data.

(10 marks)

After login, as shown in the Figure Q2(a), the Container area will display no packages.

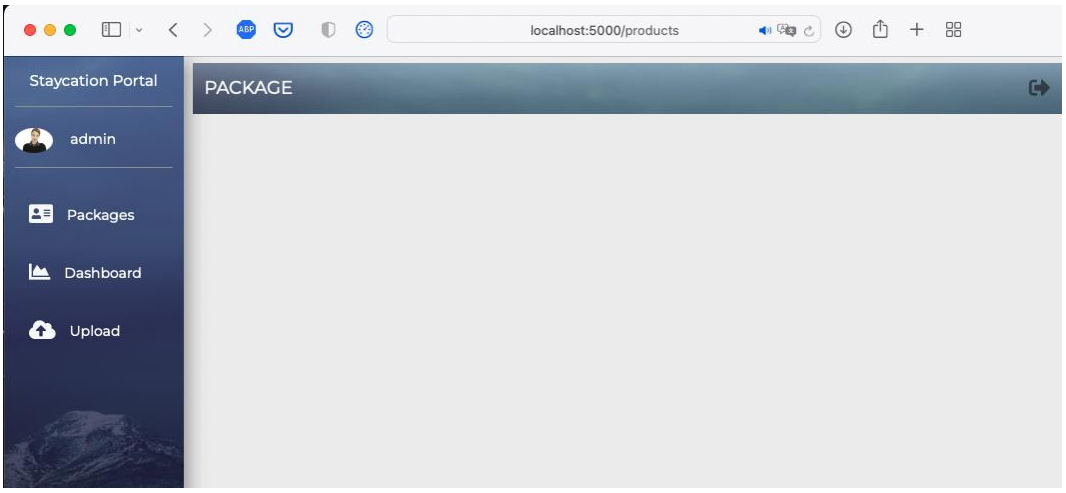

Figure Q2(a): An empty package view when the user performs a first login

(b) Develop the Upload view and Staycation and Booking Models:

As shown in the SideBar area of Figure Q2(a), when logged in, the admin user is presented with three links; the Upload link is to upload Users, Staycation and Booking records from files, the Dashboard link is to display the chart of the incoming over a period for the platform, and the Packages link is to generate the same view as shown in Figure Q1(e) but based on data stored in NoSQL database after the data files are uploaded.

(i) When the Upload link is clicked, the application will display the page as shown in Figure Q2(b). The admin user selects the data type to update, the options include Users, Staycation, or Booking as shown in Figure Q2(c).

When the Browse Button is clicked a directory browser will pop-up for the admin user to open the selected file to be upload by clicking he Upload button as shown in Figure Q2(d).

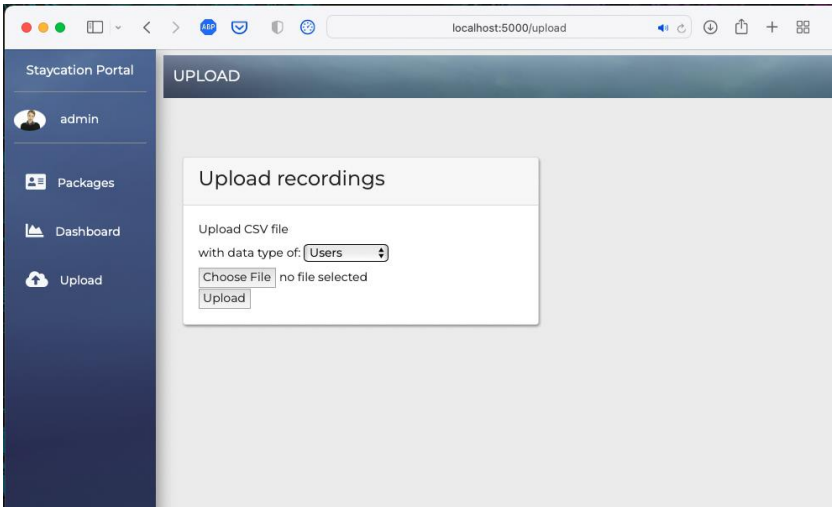

Figure Q2(b): The Upload View

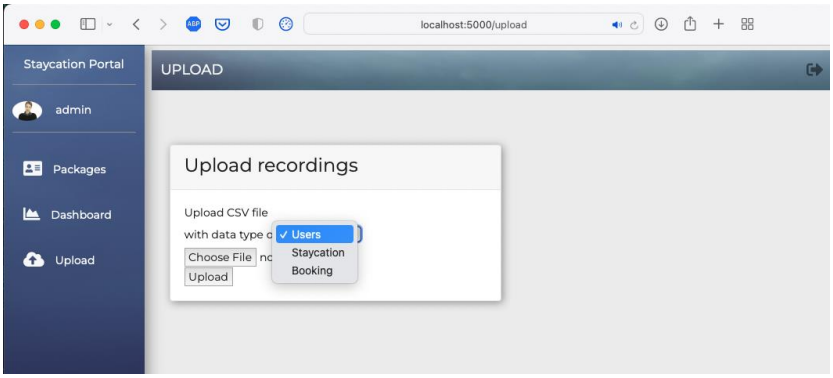

Figure Q2(c): The Selection Option

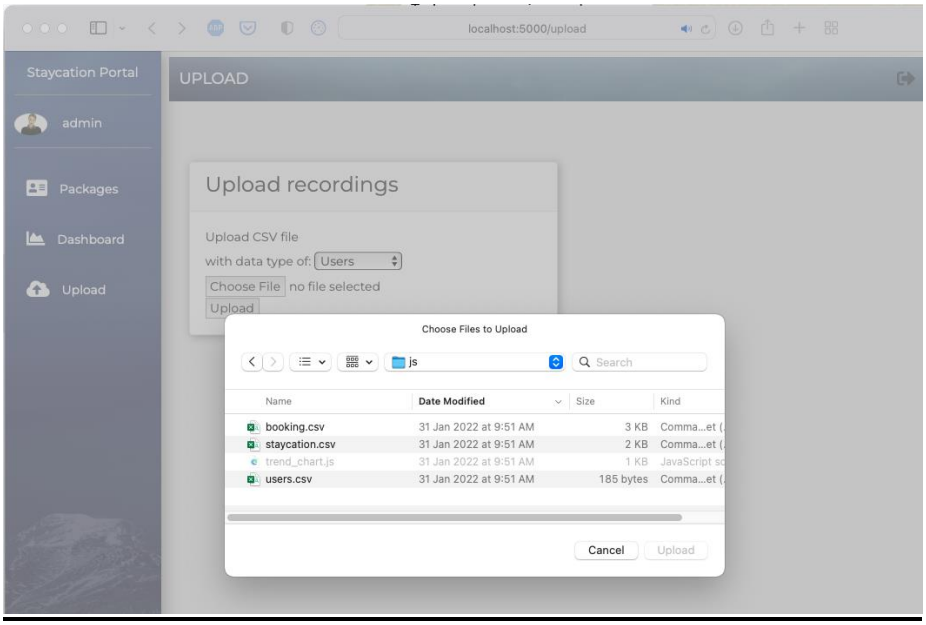

Figure Q2 (d): When Browse button is clicked

(10 marks)

(ii) The data files provided are "staycation.csv", "users.csv", and "booking.csv" to be uploaded to populate the data models for Staycation, Users and Booking respectively.

The schema for the Users model is as shown in Table Q2(a). The schema for the staycation and booking model are shown below:

Hotel_name	Duration	Unit_cost	Image_url	Description

Table Q2(b): The Schema of a Staycation based on staycation.csv

Check_in_date	Customer	Hotel_name

Table Q2(c): The Schema of a Booking based on booking.csv

Employ NoSQL database to implement the schema for the database to store the uploaded data.

(10 marks)

(c) After the data is populated in NoSQL database via the solution to Q2 (b), (re-)develop the solution to Q1 (c) to achieve the same views and business processes from data stored in the NoSQL database.

(15 marks)

(d) Develop the Dashboard view and the Chart Model.

When the admin user click on the Dashboard link a chart that indicate the total income from the bookings of a period of time will be displayed as shown in Figure Q2 (e).

The chart display the total daily booking income ranging from 17 of January 2022 to 12 March of 2022 for each hotel, indexed by hotel name that offers a staycation package.

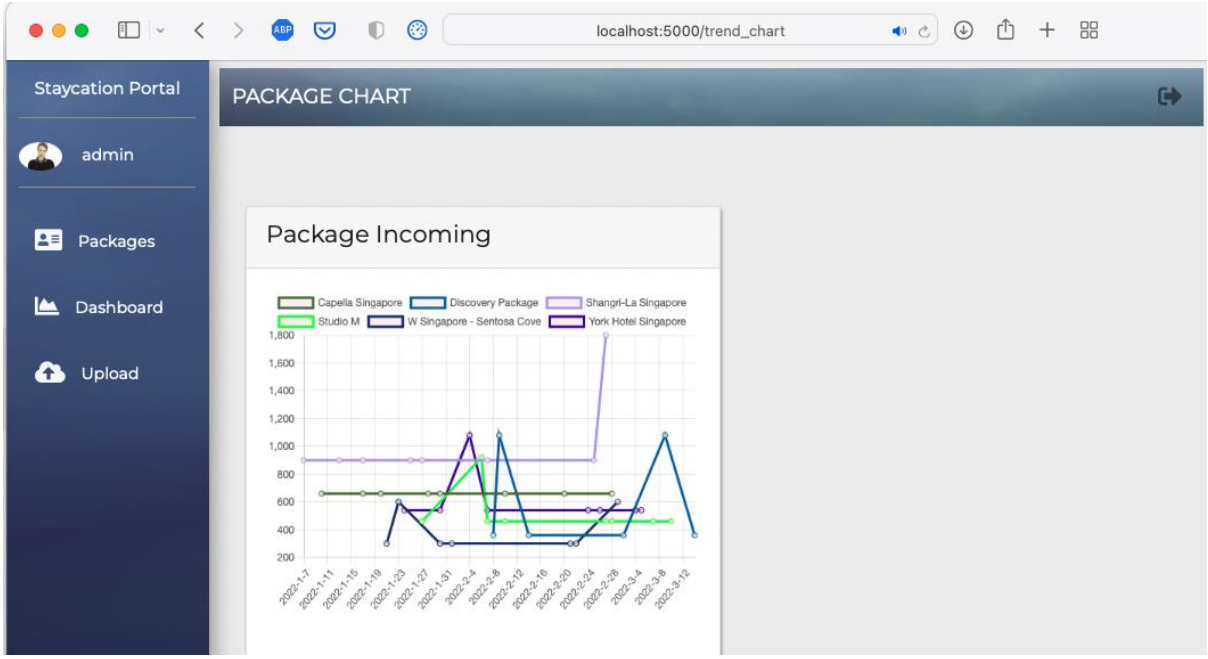

Figure Q2 (e): Dashboard showing the Total Incoming of Hotels over a period

(10 marks)

---- END OF ASSIGNMENT ----