**WEB APPLICATION FRAMEWORK**

Name: Van Phi Vu Huynh

Student Id: 20141439

Tutor: Tristan Reed

Task Questions

1. Despite whichever way you went about it, list the components that you have built in the frontend and whether or not they could be implemented as a functional (rather than class-based) component. If this is not possible, state why the component can’t be a functional component (this may relate to a parent or child component).

Components that I built in this assignment:

* AddFlight.js
* FlightCard.js
* MainPage.js

These components can’t be a functional component since I defined other functions likes get data, edit data, or format display data. Functional components are used to display data from props only.

1. Compare and contrast the two approaches to web application development which were discussed in this unit: the ‘traditional’ (Django style) approach and single view/progressive web applications (Express.js and React.js). Provide at least three examples to support your argument.

Traditional approach:

* The whole page will load again when we switch to another link,
* It loads required page only.
* User experience may not be good if the network is slow.
* Can be built by many coding languages (support by huge number of framework).
* Slow at development.

Single view approach:

* The whole page won’t load again switch to another link. It just renders a new component.
* It loads all components in the page even we may not navigate to it.
* User can have a good user experience once the webpage has done loaded.
* Built on JavaScript only, may not work if user prevent from loading scripts in the browser.
* Fast at development.

1. Do you think it would be appropriate to use the Django admin interface for users to create, update and delete data on a Django web application? Support your argument with at least two different reasons.

No.

Because the administrative interface allows the user to perform all actions to change the data in the database, including all important data such as the administrative user account. When they have that privilege, they can sabotage the company's business.

1. Why should the response to a POST request redirect the user to another web page, rather than serve the web page directly back to the user? (Hint: consider other types of REST requests).
2. Express.js and Django (with a plugin such as Tastypie) both allow the creation of a backend API which a frontend such as that built in React could be implemented within. If the client interface was served via a Django ‘view function’, state a downside in terms of deployment within an AWS architecture, when compared to an ‘independent’ client interface (i.e. how we do it with Express.js and React.js together).

Since Django view function will reload the page when user make a request or navigate to another menu. This will increase the bandwidth usage which cost more money for AWS.

1. How would you build the data model if this was a Django project? State (in terms of what models.py would contain) what it would look like if this project was to be implemented in Django.

Models:

Text

Description automatically generated

If I can build this project in Django, I could use the advantage of built-in model to support me in during development. It helps me in creating a new plane easier, like I can load a list of cities, airplanes, etc, to a dropdown list where I can choose from without type it in.