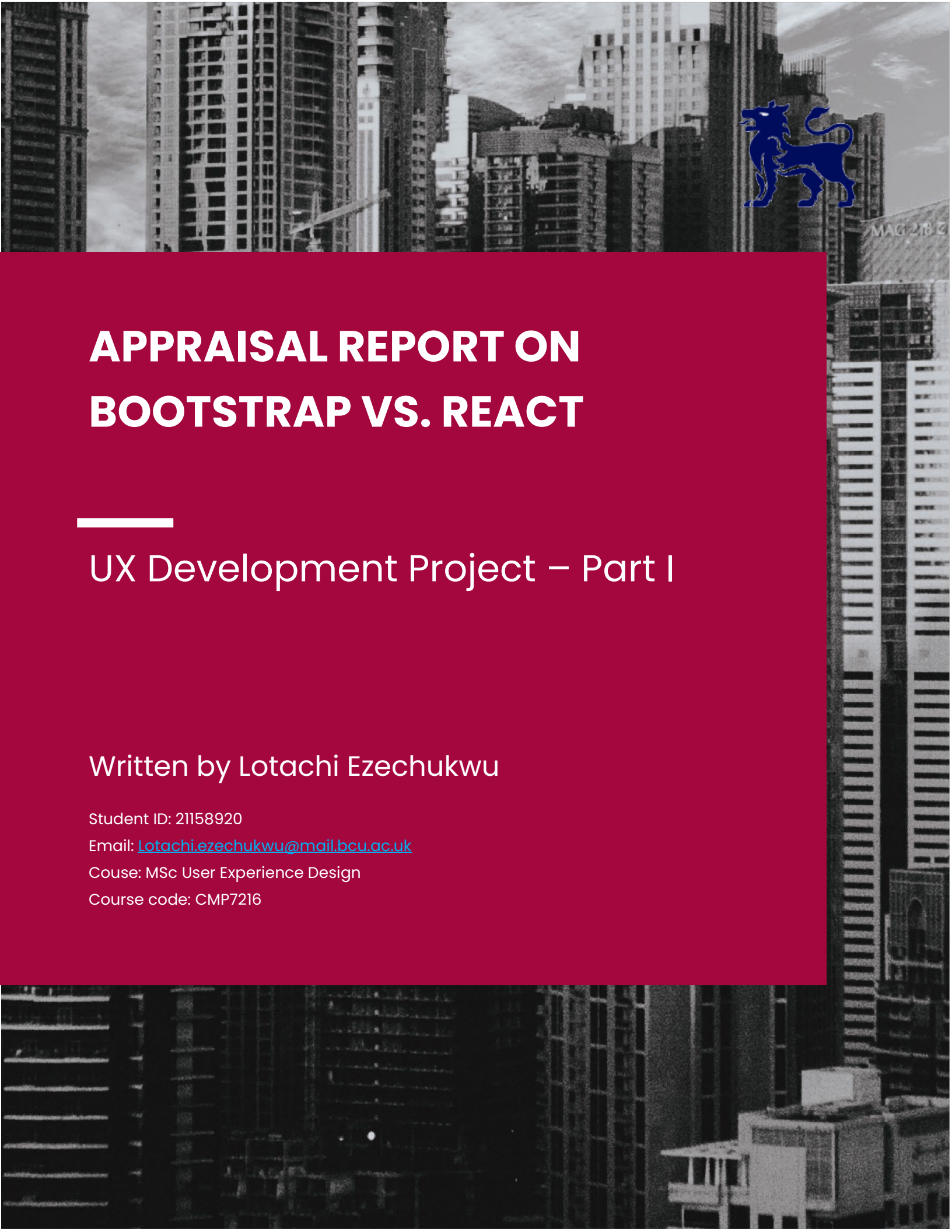

APPRAISAL REPORT ON BOOTSTRAP VS. REACT

UX Development Project – Part I

Written by Lotachi Ezechukwu

Student ID: 21158920

Email: Lotachi.ezechukwu@mail.bcu.ac.uk

Couse: MSc User Experience Design

Course code: CMP7216

The Brief: Evaluate 2 different front-end frameworks

Front-end frameworks are collections of HTML, CSS, and JavaScript code that developers use to build web applications. There are many options to choose from, including *AngularJS, React, Vue.js, Ember.js, Foundation, Bootstrap, Semantic UI, Bulma, Materialize and Tailwind CSS.* This is not an exhaustive list, as new frameworks are constantly being developed.

The choice of which framework to use depends on the specific needs and goals of the project, as well as the preferences of the development team. In this report, we will focus on *Bootstrap* and *React.*

BOOTSTRAP

Bootstrap is a front-end framework. It is not a programming language, but rather a collection of HTML, CSS, and JavaScript code that developers use as a starting point to build web applications.

Bootstrap was originally developed by Twitter to standardize the design of its internal tools. It was created by a small team of designers and developers at Twitter, led by Mark Otto and Jacob Thornton.

The first version of Bootstrap was released in 2011, and quickly gained popularity among developers to build responsive, mobile-first websites and web applications. It was designed to be easy to use and customize and provided a set of predefined styles and layout components that could be used as a starting point for building user interfaces.

Over the years, Bootstrap has evolved, with new versions, becoming one of the most popular front-end frameworks with millions of websites and web applications built using it. It is now maintained by a large open-source community of developers and designers and is available for anyone to use and contribute to.

Its widespread popularity is evident in its inclusion as a default in many web development templates and starter kits.

Bootstrap can be used in a few ways to build web applications. Here are a few common ways developers use:

1. **As a base for a custom design:** Its predefined styles and layout components can be used as a foundation for a custom design, and then customized with own CSS for a quick creation of a professional-looking application without starting from scratch.
2. **As a set of ready-made components:** Developers can use its predefined components to add common elements such as buttons, forms, and navigation menus, saving time and effort.
3. **To ensure cross-browser compatibility :** This is for consistent application appearance and function across platforms.

To use Bootstrap, developers will typically include the Bootstrap CSS and JavaScript files in their application, and then use the predefined styles and layout classes in their HTML code. They can also customize the appearance of their application by overriding the default Bootstrap styles with their own CSS.

Bootstrap can be used to create a wide variety of interactions in user experience (UX) design. Some examples include:

1. **Navigation:** Bootstrap provides various components for creating navigation menus, such as navbars, dropdown menus, and breadcrumb trails.
2. **Forms:** Bootstrap includes styles for form elements such as input fields, buttons, and checkboxes, as well as styles for error messages and form validation.
3. **Modals:** They are dialog boxes used to display content or gather input from the user.
4. **Tooltips and popovers:** They provide additional information or context to the user.
5. **Carousels:** They are sliders that can be used to display a series of images or other content.

Bootstrap is not a library. It is a framework, which is a comprehensive set of tools that provides a structure for building a complete application. In contrast, a library is a collection of pre-written code that developers can use to add specific functionality to their projects. A library typically only offers specific functions or utilities that can be called upon as needed. For example, the jQuery library is a collection of utility functions for manipulating the DOM, making HTTP requests, and handling events, but it does not include layout tools or pre-designed UI components like Bootstrap does.

REACT

React is a JavaScript library for building user interfaces. It was created by Jordan Walke, a software engineer at Facebook, who released the first version in May 2013.

At the time, Facebook was building a large, complex web application called Facebook Ads Manager, and the team was struggling to manage the increasing complexity of the application's UI. In response, Walke developed a new approach to building user interfaces that focused on breaking down the UI into reusable components and using a virtual DOM to optimize the rendering of these components.

The idea behind React was to make it easier to build and maintain large, dynamic web applications by allowing developers to declaratively describe the UI as a function of the application's state. This approach, known as declarative programming, made it easier for developers to reason about the UI and made it easier to build and modify the UI as the application evolved.

React uses a virtual DOM (Document Object Model) to optimize the rendering of UI elements, which makes it faster and more efficient than other approaches that manipulate the DOM directly.

To use React, you need to include the React library in your project and write your UI components as a series of functions or classes that return JSX (JavaScript XML) elements.

"JSX is a syntax extension for JavaScript that allows you to write HTML-like code in your React components." (*"The Essential Guide to Building Web Applications with React")*

When your components are rendered, the JSX is compiled into vanilla JavaScript that manipulates the DOM to update the UI.

React is often used to build single-page applications (SPAs) and mobile-first web applications, but it can be used in any type of web project. It is not a mobile development framework and not designed to build mobile apps. But mobile apps can be built using a framework called React Native. Although React and React Native are two different technologies, they were both developed by Facebook for building user interfaces.

React Native is a mobile development framework that was released in 2015. ("Cost-Effectiveness of React Native for Mobile App Development") It allows developers to build native mobile apps using a variant of the React syntax. Instead of rendering to the DOM like React does, React Native translates the component hierarchy into native platform-specific components, such as iOS or Android views. This allows developers to build apps that have a native look and feel and that can access native platform APIs, such as the camera or the accelerometer.

In summary, React is a library for building web user interfaces, while React Native is a framework for building native mobile apps. ("Difference Between React.JS and React Native - Codersid") Both technologies use the same syntax and core concepts, but they are designed for different platforms and have different capabilities.

React can be used to create a wide variety of interactions in user experience (UX) design. Some examples include:

1. **Navigation**: React is used to build dynamic navigation menus that change in response to user actions or application state.

2. **Forms:** React is used to build forms with real-time validation and error handling. It is also used to build custom form controls, such as date pickers or colour pickers.
3. **Modals:** React is used to build modals that are triggered by user actions or that are displayed in response to certain application events.
4. **Tooltips and popovers:** React is used to build custom tooltips and popovers that provide additional information or context to the user.
5. **Animations:** React is used to build custom animations that add visual interest and interactivity to the UI.

Its powerful component model and declarative syntax make it a very useful tool for building dynamic and interactive user interfaces.

Since its release, React has become a very popular JavaScript library for building user interfaces. According to the 2020 Stack Overflow Developer Survey, React was the most popular web framework among professional developers, with 63.5% of respondents saying that they had used it in the past year.

There are several reasons for React's popularity among developers.

- Easy to learn and use, especially for developers familiar with JavaScript, with declarative syntax and reusable components for building and maintaining complex UIs.
- Its simplicity, flexibility, and performance benefits, including use of virtual DOM, which makes it faster and more efficient than other approaches that manipulate the DOM directly, making it suitable for large, dynamic web applications.
- Inspired other technologies like React Native, allowing for native mobile app development using a variant of the React syntax.
- React has a large, active developer community and many available resources for learning and finding support, as well as open-source libraries and tools built using React.

BOOTSTRAP VS. REACT – THE CONCLUSION

Bootstrap and React are two different technologies that are often used together in web development, but they serve different purposes.

Bootstrap is a front-end framework, developed by Twitter, with pre-designed and pre-styled UI components, including styles for common web elements such as buttons, forms, and tables, as well as layout tools such as a grid system and responsive design features.

React is a JavaScript library for building user interfaces, developed by Facebook, with a virtual DOM for efficient rendering of reusable UI elements.

In a web project, Bootstrap and React can be used together to build a user interface. For example, developers can use Bootstrap's pre-designed UI components and layout tools to quickly build a basic layout, and then use React to add dynamic behaviour and interactivity to the UI.

However, React and Bootstrap can be used independently of each other. The choice of which technologies to use will depend on the specific needs and goals of the project.