### Technical Report for Church App

### (19 July 2024)

### (Week 2)

#### . Project Overview

The Church App 2 is a web-based application designed to facilitate various activities related to church events, prayer requests, online shopping, and more. This app provides users with an intuitive interface to interact with the church community and participate in events and activities.

#### Dependencies

The project utilizes several npm packages, as listed in the package.json file. Here are some key dependencies:

* **Express**: Used for setting up the server and handling HTTP requests.
* **Nodemailer**: Used for sending emails.
* **Other utility libraries**: Various other libraries for handling application logic.

#### . Key Features

1. **User Authentication**: Pages like Login.html and Signup.html provide forms for user authentication.
2. **Home Page**: The Home.html file serves as the landing page for the application.
3. **Event Management**: Events.html and Events.css handle the display and styling of church events.
4. **Prayer Requests**: The PrayerRequest.html page allows users to submit prayer requests.
5. **Online Shop**: Shop.html, Shop.css, and shop.js manage the online shop functionality.
6. **Chatbot**: The Chatbot.html page implements a chatbot for user interaction.

### Detailed File Descriptions

* **app.js**: The main server-side JavaScript file that sets up the Express server and handles routing. It initializes the server, sets up middleware, defines routes, and starts listening on a specified port.
* **Chatbot.html**: Implements a chatbot interface for user interaction. This HTML file includes the structure and elements necessary to create a chatbot interface, potentially integrating with a chatbot service or custom chatbot logic.
* **Events.css**: Contains the styles for the events page. It defines the visual presentation of event-related elements, such as event listings, descriptions, and other stylistic components to ensure a consistent and appealing look.
* **Events.html**: Displays a list of church events. This HTML file provides the structure and layout for showing upcoming and past events, including details like date, time, location, and descriptions.
* **Home.html**: The main landing page of the application. It serves as the central hub for users when they first visit the site, often including links to other parts of the app, a welcome message, and other relevant information.
* **Login.html**: Provides a login form for users. This HTML file includes input fields for username and password, as well as buttons for submitting the login form and possibly links to password recovery or sign-up pages.
* **package.json**: Lists the project's dependencies and metadata. It includes information like the project name, version, description, author, license, and scripts. It also specifies the npm packages required to run the application.
* **package-lock.json**: Records the exact versions of npm dependencies. It ensures that the project is built with the same dependency versions across different environments by locking the specific versions of all installed packages.
* **Prayer/**: Directory presumably containing resources related to prayer features. This may include additional HTML files, scripts, styles, or server-side code specific to handling prayer requests, prayer groups, or related functionality.
* **PrayerRequest.html**: Allows users to submit prayer requests. This HTML file provides a form where users can enter their prayer requests, which are then submitted to the server for processing or display.
* **script.js**: General JavaScript file for handling client-side logic. It may contain functions and event handlers for various interactive elements across the application, enhancing the user experience with dynamic behaviours.
* **Shop.css**: Styles for the shop page. This CSS file defines the visual appearance of the online shop, including the layout of product listings, styling of buttons, and overall theme of the shopping section.
* **Shop.html**: Displays the online shop. This HTML file structures the content for the shop, including product listings, descriptions, prices, and potentially a shopping cart interface.
* **shop.js**: JavaScript file for handling shop-related functionality. It may include code for managing the shopping cart, processing payments, updating product quantities, and handling user interactions within the shop.
* **Signup.html**: Provides a signup form for new users. This HTML file includes fields for collecting user information such as name, email, password, and other relevant details required for creating a new account.
* **styles.css**: Contains general styles for the application. This CSS file defines the global styles used throughout the app, ensuring a consistent look and feel across all pages and components. It may include basic layout settings, typography, color schemes, and other foundational styles

#### Database Integration

The application uses an SQL database to store user information, events, prayer requests, and shop data. The specific SQL database used would be configured in the application settings. Integration with the SQL database is managed through direct SQL queries.