**Application Report:**

Prepared by: Jiaqi Guo (Max)

Contents

[Application Experience: 2](#_Toc40523350)

[File Structure Diagram 4](#_Toc40523351)

[File Structure Expatiation 5](#_Toc40523352)

[Server.js： 5](#_Toc40523353)

[Config folder: 5](#_Toc40523354)

[Config.json: 5](#_Toc40523355)

[Models folder: 5](#_Toc40523356)

[Index.js 5](#_Toc40523357)

[User.js 5](#_Toc40523358)

[Public folder: 5](#_Toc40523359)

[Login.html 5](#_Toc40523360)

[Signup.html 6](#_Toc40523361)

[Member.html 6](#_Toc40523362)

[Js folder 6](#_Toc40523363)

[Stylesheets folder 6](#_Toc40523364)

[routes folder 6](#_Toc40523365)

[api-routes.js 6](#_Toc40523366)

[html-routes.js 6](#_Toc40523367)

[Package.json 6](#_Toc40523368)

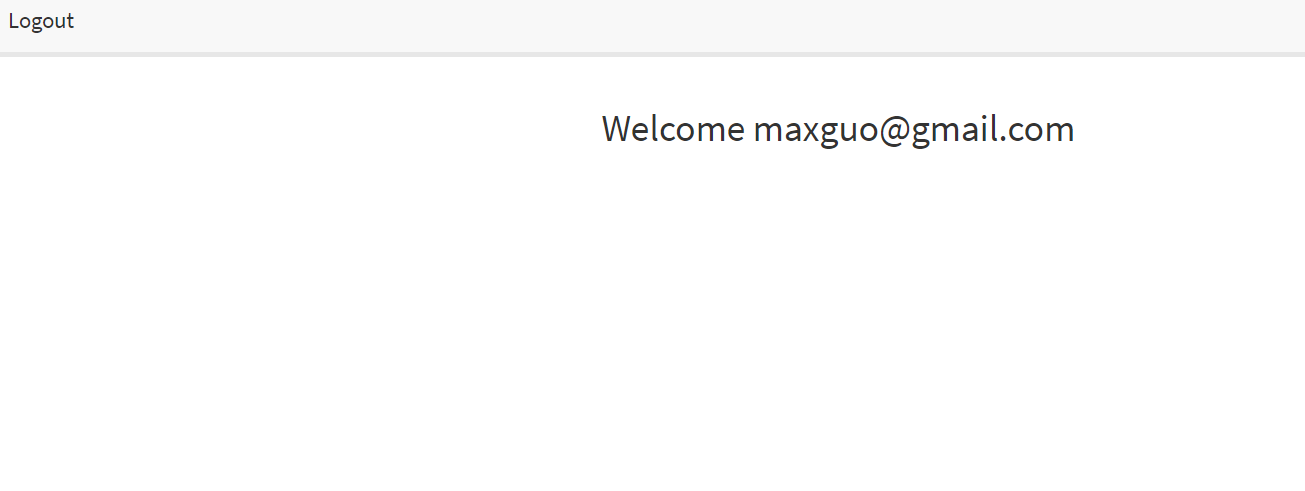
[Node\_moduels 6](#_Toc40523369)

# Application Experience:

After application starts, open Browser and go to home page. The signup page will appear as default. This page is handled by signup.html file from the public folder. User could put in email address and password to create an account. After click ‘sign up’ user will create an account and log in right away.

After login the user email will display from the page, this part is handled by the member.html from the public folder.





If user already has an account. Click on the blue here button to login.



File Structure Diagram

# File Structure Expatiation

## Server.js：

Server.js file is the application’s entry point, by running this file, application starts the server, initiate the port listening and establish the route methods to the different http requests.

The client-side files are stored in the public directory, the route method files are stored in the route’s directory.

## Config folder:

Config folder stores information about database and method to handle authentication’s results consequence.

### Config.json:

Config.json stores database connection information, this information will be used by the npm sequelize module

#### Middleware:

Middleware folder stores all the middleware method for server.

#### isAuthenticated.js

isAuthenticated.js stores method to handle consequence of the authentication result. If pass authentication, proceed to restricted routes, else return to the root login page.

## Models folder:

Models folder stores all the database models. Database models represent tables from the database.

### Index.js

Index.js file encapsulates all the models from the model’s folder and establish certain relationships between models if exists.

### User.js

User.js file stores the User model, user model has two fields from the database, email field and password field. There will be a encrypt method applies to the database to enhance security.

## Public folder:

public folder stores all the files to handle client-side content, including user interface, methods and styling.

### Login.html

Login.html provides interface to let user login or sign up. The sign-up option will navigate user to the signup.html page.

### Signup.html

Signup.html provides form to let user enter information to create an account for the website and the information will be stored in the database.

### Member.html

Member.html stores information when user successfully login to the web application. Currently there are not much logic here, this section could be widely expanded.

### Js folder

Js folder stores all the backend methods used from the client-side

#### Login.js

Stores login method for login.html page

#### Signup.js

Stores signup method for signup.html page

#### Member.js

Stores member page logic, currently there are not much: return and display login user’s email.

### Stylesheets folder

stylesheets folder stores CSS styling files for the client-side interface

#### style.css

stores client-side styling

## routes folder

routes folder stores all the http request handler files

### api-routes.js

stores method to handle http api request

### html-routes.js

stores method to handle http html request

## Package.json

Package.json file is a standard npm file stores all the modules will be used for the application

## Node\_moduels

Node\_modules folder stores all the modules used for the application.