

# CSCC01 eBeauty Project Documentation

Lim Lei,  
Allen Chien-Luan Lo,  
Jason Chen,  
Jin Ting Zhou,  
Junxun Liu,  
Hyun Woo (Eddie) Shin,  
Yuehai Zhou

July 30, 2022

## Table of Contents

<b>Frontend.....</b>	<b>5</b>
src/actions/providerAction.js .....	5
src/Authentication/AuthProvider.js .....	5
src/Authentication/RequireAuth.js.....	5
src/Authentication/useAuth.js .....	5
src/components/AddressForm.js.....	5
src/components/Alerting.js.....	5
src/components/Animation.js .....	5
src/components/CustomerCalendar.js .....	5
src/components/CustomerNavbar.js .....	5
src/components/Footer.js.....	6
src/components/Loader.js .....	6
src/components/MapContainer.js.....	6
src/components/Message.js .....	6
src/components/Navbar.js.....	6
src/components/Paginator.js .....	6
src/components/ProviderCalendar.js .....	6
src/components/ProfileCard.js.....	6
src/components/ProviderNavbar.js.....	6
src/components/SampleGallery.js.....	7
src/components/ScrollToTop.js .....	7
src/components/SearchBox.js .....	7
src/components/ServiceDropdown.js.....	7
src/constants/countries.js.....	7
src/constants/creditCardmonths.js.....	7
src/constants/creditCardYears.js.....	7
src/constants/providerConstant.js .....	7
src/constants/provinces.js .....	7
src/pages/AddressPage.js .....	8
src/pages/CheckoutAddressPage.js .....	8
src/pages/CheckoutPaymentPage.js.....	8
src/pages/CheckoutPaypalPage.js .....	8

src/pages/CheckoutReviewPage.js .....	8
src/pages/ContactPage.js .....	8
src/pages/FindJob.js .....	8
src/pages/HomePage.js .....	8
src/pages/Login.js .....	8
src/pages/NotFound.js .....	9
src/pages/OnlyProviderView.js .....	9
src/pages/OrderHistoryPage.js .....	9
src/pages/ProfileProvider.js .....	9
src/pages/ProviderScheduling.js .....	9
src/pages/ReservationCustomer.js .....	9
src/pages/ReviewPage.js .....	9
src/pages/SearchPage.js .....	9
src/pages/ServiceMenuPage.js .....	9
src/pages/SignUp.js .....	10
src/pages/SignUpProvider.js .....	10
src/pages/SignUpProviderTwo.js .....	10
src/pages/SignUpProviderThree.js .....	10
src/pages/Unauthorized.js .....	10
src/pages/ViewProfileProvider.js .....	10
src/pages/OrderHistory.js .....	10
src/reducers/providerReducer.js .....	10
src/App.js & src/index.js .....	11
src/store.js .....	11
<b>Backend .....</b>	<b>11</b>
/backend/config/db.js .....	11
/backend/controllers/calendarController.js .....	11
/backend/controllers/customerController.js .....	11
/backend/controllers/issueController.js .....	11
/backend/controllers/orderController.js .....	11
/backend/controllers/postController.js .....	11
/backend/controllers/providerController.js .....	12
/backend/controllers/reviewController.js .....	12

/backend/data/metrics.json .....	12
/backend/data/order.json .....	12
/backend/data/post.json .....	12
/backend/data/profiles.js .....	12
/backend/data/providers.js .....	12
/backend/data/review.js .....	12
/backend/data/service.js .....	12
/backend/middleware/authMiddleware.js .....	12
/backend/middleware/errorMiddleware.js .....	13
/backend/middleware/upload.js .....	13
/backend/models/calendarModel.js .....	13
/backend/models/customerModel.js .....	13
/backend/models/orderModel.js .....	13
/backend/models/issueModel.js .....	13
/backend/models/postModel.js .....	13
/backend/models/providerModel.js .....	13
/backend/models/reviewModel.js .....	13
/backend/models/serviceModel.js .....	13
/backend/routes/calendarRoute.js .....	14
/backend/routes/customerRoute.js .....	14
/backend/routes/imageRoute.js .....	14
/backend/routes/issueRoute.js .....	14
/backend/routes/orderRoute.js .....	14
/backend/routes/postRoute.js .....	14
/backend/routes/providerRoute.js .....	14
/backend/routes/reviewRoute.js .....	14
/backend/Server.js .....	15
/backend/Seeder.js .....	15

## Frontend

### `src/actions/providerAction.js`

This is a file containing all the actions of provider, it fetched data from backend to the frontend to get the list of providers or detail of a provider

### `src/Authentication/AuthProvider.js`

it warps and binds the context, making react-context accessible for all children under it.

### `src/Authentication/RequireAuth.js`

It's to restrict some specific (defined in App.js) pages to be only accessible for a certain identity. For example, only providers can access their own schedule, but if a customer wants to access that url, they would be blocked.

### `src/Authentication/useAuth.js`

It's a hook of useContext, every class can import it and access the context value. Critical use of it is to specify if logged in identity can access specific pages.

### `src/components/AddressForm.js`

This file contains a form that saves the user's input of the address.

### `src/components/Alerting.js`

This file is a helper function that implement the react-notification-component and used to replaced the normal alert();

### `src/components/Animation.js`

This file is under FindJob.js that it provides the animation of fading and appearing images in that page.

### `src/components/CustomerCalendar.js`

The file is used to provide a calendar which the provider has, the customer should and can look into it and see if the provider is busy.

### `src/components/CustomerNavbar.js`

It's the navbar user has after logged in, not fully implemented yet.

#### src/components/Footer.js

This file gives a footer at the bottom of our website to have access to other info of the website.

#### src/components/Loader.js

This file is a loader component when a user is trying to get data from DB, it will show a spinner on the screen.

#### src/components/MapContainer.js

This file links to google map api to produce a map that shows the location of the user's input address.

#### src/components/Message.js

This file is a message component when something is wrong, and it will show the error message on the website, currently only used for search page results.

#### src/components/Navbar.js

This file gives a navbar at the top of our website. It includes the title of our website, searching of services and links to other pages including Login.js, SignUp.js, etc.

#### src/components/Paginator.js

This file provides a pagination component when results show up in the search page, and users can navigate to different page for more results

#### src/components/ProviderCalendar.js

It's the calendar that only providers can access to, and they would be able to look at a reservation detail, add/delete them when they want to do so.

#### src/components/ProfileCard.js

This a card component that contains the rough data of task providers when users search on the search page

#### src/components/ProviderNavbar.js

it's the navbar provider that has after logged in, currently it only links to the provider schedule page and logout functionality.

#### [src/components/SampleGallery.js](#)

The file is a component used to display multiple images of sample works from service providers in their profile pages.

#### [src/components/ScrollToTop.js](#)

This file is just a helper component that wraps contents inside the App.js. It is used for resetting the client's page position to the top after switching pages.

#### [src/components/SearchBox.js](#)

This file provides a search bar component that users can type keywords when they want search

#### [src/components/ServiceDropdown.js](#)

This file contains the component responsible for providing the list of services available for a service receiver to choose from, on the search page.

#### [src/constants/countries.js](#)

This file contains a constant hashmap of different countries used in CheckoutAddressPage.js

#### [src/constants/creditCardmonths.js](#)

This file contains a constant hashmap of credit card months data which is used in CheckoutPaymentPage.js for credit card expiry info.

#### [src/constants/creditCardYears.js](#)

This file contains a constant hashmap of credit card years data which is used in CheckoutPaymentPage.js for credit card expiry info.

#### [src/constants/providerConstant.js](#)

This file contains all the constants for provider route, which include the constants of states.

#### [src/constants/provinces.js](#)

This file contains a constant hashmap of province data in Canada.

#### [src/pages/AddressPage.js](#)

This file puts the following two files together with Footer.js together to form the address page that allows users to enter their new address.

#### [src/pages/CheckoutAddressPage.js](#)

This file provides an input form for customers to provide their addresses. The address provided in this page can be used for service providers or billing purposes.

#### [src/pages/CheckoutPaymentPage.js](#)

This file provides an input form for credit card information. Customers can provide their credit card information for payment.

#### [src/pages/CheckoutPaypalPage.js](#)

This file displays checkout data for customers to review and allows customers to use their paypal accounts to make payment by enabling paypal access buttons.

#### [src/pages/CheckoutReviewPage.js](#)

This file provides an overview of the order which is about to be submitted, and customers can review to check whether their information is correct.

#### [src/pages/ContactPage.js](#)

This file provides a place for the users to find the contact information of eBeauty, along with a webform allowing them to contact support when they need help/additional information.

#### [src/pages/FindJob.js](#)

This file provides a webpage where the potential service provider can access this page through the navbar at the top of the home page, and register as an official service provider on Amorr.

#### [src/pages/HomePage.js](#)

This file is used as the main page for the website, and showcases some of the features to the users.

#### [src/pages/Login.js](#)

This file allows existing users and providers to login to their account.



[src/pages/NotFound.js](#)

This file just uses a link to redirect users back to the homepage (Home.js) if they input an invalid url.

[src/pages/OnlyProviderView.js](#)

It's a testing page that only the provider can access to, will delete this in future.

[src/pages/OrderHistoryPage.js](#)

This page allows users to view their past and upcoming orders.

[src/pages/ProfileProvider.js](#)

This file provides the service provider a place to edit his/her profile or account information including names, address, phone number, email or password.

[src/pages/ProviderScheduling.js](#)

It's a page that retrieves calendar info of the provider using it, this file uses ProviderCalendar as a component, and manipulates it by extra user interface.

[src/pages/ReservationCustiommer.js](#)

The file provides the customer to view a specific appointment/reservation they have made or already finished.

[src/pages/ReviewPage.js](#)

This file is used to collect customer reviews for their services received by service providers with star rating from 1 to 5. The collected data are stored in Review DB.

[src/pages/SearchPage.js](#)

This file is the component for a search result page, it should show the user's current address on file and allow users to re-search the service if they are not satisfied with the current result. It also shows a categorical dropdown, where users can modify and filter those categories. And the result of service providers.

[src/pages/ServiceMenuPage.js](#)

This file contains all available services from service providers for reservation, and displays a calendar with service providers' availability so that customers can choose the service they want to book and select the available date and time.

`src/pages/SignUp.js`

This file allows users without an account to sign up (just for non-service providers).

`src/pages/SignUpProvider.js`

This file provides a place for a person to enter his/her basic info to set up an account (first step). Then redirect to the second step of registering an account.

`src/pages/SignUpProviderTwo.js`

This file provides a place for a person to select and apply his/her conditions and what to serve and set the range for his/her service to set up an account (second and maybe the last step).

`src/pages/SignUpProviderThree.js`

This file provides a place for the provider to upload a profile image during registration. and then post the image and provider info to the database separately. Right now it redirects back to the homepage Home.js.

`src/pages/Unauthorized.js`

It's supposed to be shown if a person wants to access a page but not allowed to, but currently it has not been implemented yet.

`src/pages/ViewProfileProvider.js`

This file provides the service receivers a place to view the profile of a specific service provider including names, address and services provided.

`src/pages/OrderHistory.js`

The file provides the customer a place to access their past or upcoming orders, and may change them by going into the reservation page.

`src/reducers/providerReducer.js`

This file is used to request reducers for all providers and a single provider. It returns loading state status and provider action payload if success.

src/App.js & src/index.js

They are used to manage and warp components of other js files. The usage is to be the root and run the website.

src/store.js

This is a store that stores all the reducers and exports them to a root reducer.

## Backend

/backend/config/db.js

The file is used to connect to MongoDB. It uses the url from .env. Shows error when there is one. It receives requests from the server, handles query forms and sends the response with data to the server.

/backend/controllers/calendarController.js

It contains the functions to create a new timeslot calendar , delete a timeslot calendar, update data, and retrieve the calendar in many different ways, like get all calendar, get calendar from one provider and get specific calendar's details from DB.

/backend/controllers/customerController.js

It contains the functions to create a new customer entry, delete a customer entry, update customer entry data or retrieve customer data from DB.

/backend/controllers/issueController.js

add the issue in to DB

/backend/controllers/orderController.js

It is used to handle the data collected from checkout pages to save into DB. For this sprint, it only saves orders into DB, but for the next sprint, the service providers can retrieve their orders for review.

/backend/controllers/postController.js

It is used to let providers post their work and modify the posts they've made. The functions inside are getting all the posts, creating a post, modifying a post, and deleting a post. (not implemented in frontend yet)

`/backend/controllers/providerController.js`

It is used to let providers register (with hashed password), login, and two apis of GET method, one to get all provider data, the other one to get a single provider details by its id.

`/backend/controllers/reviewController.js`

It contains the functions to create a new customer review entry, delete a customer review, update customer review data or retrieve customer review data from DB.

`/backend/data/metrics.json`

sample data of provider data dashboard

`/backend/data/order.json`

sample data of provider data dashboard

`/backend/data/post.json`

sample data of provider data dashboard

`/backend/data/profiles.js`

sample data of profile schema

`/backend/data/providers.js`

sample data of providers' schema

`/backend/data/review.js`

sample data of review schema

`/backend/data/service.js`

sample data of service schema

`/backend/middleware/authMiddleware.js`

This file usage is related to posts, only authenticated providers can make modifications to the posts. it also

`/backend/middleware/errorMiddleware.js`

This file is used to handle errors and display them properly. One example is that if we faced any error, it can handle the error and show it inside postman's response block

`/backend/middleware/upload.js`

It is a helper middleware to help generate multer & bucket that is used for GRIDFS. then the help function can then post the image uploaded to the database.

`/backend/models/calendarModel.js`

schema of calendar for mongoDB

`/backend/models/customerModel.js`

schema of customer for mongoDB

`/backend/models/orderModel.js`

schema of order for mongoDB

`/backend/models/issueModel.js`

schema of issue for mongoDB

`/backend/models/postModel.js`

schema of post for mongoDB

`/backend/models/providerModel.js`

schema of provider for mongoDB

`/backend/models/reviewModel.js`

schema of review for mongoDB

`/backend/models/serviceModel.js`

schema of service for mongoDB

#### `/backend/routes/calendarRoute.js`

It defines the endpoints of Calendar DB query activities. Each route defined in the file is responsible for different tasks from getting calendar data in various different ways, and post, update, delete a calendar from the DB.

#### `/backend/routes/customerRoute.js`

It defines the endpoints of Customer DB query activities. Each route defined in the file is responsible for different tasks from letting users log in, signing up new users to getting users' personal data and addresses.

#### `/backend/routes/imageRoute.js`

This file is used to directly upload images like controllers, for the post method specifically, it uses `/middleware/upload` to generate a bucket and post the file, then returns the filename and url to access the file. (in future we'll separate it to route and controller)

#### `/backend/routes/issueRoute.js`

Routing post issue properly and directs to the issueController.

#### `/backend/routes/orderRoute.js`

It defines the endpoints of order DB query activities. routing the methods to functions in orderController

#### `/backend/routes/postRoute.js`

It defines the endpoints of post DB query activities. routing the methods to functions in postController

#### `/backend/routes/providerRoute.js`

It defines the endpoints of provider DB query activities. routing the methods to functions in providerRoute

#### `/backend/routes/reviewRoute.js`

It defines the endpoints of Review DB query activities. Each route defined in the file is responsible for different tasks from creating new customer reviews, retrieving reviews, deleting reviews and updating reviews.

`/backend/Server.js`

The file is used to initialize an Express object which can be used as a server-side pointer to communicate with both the frontend components and the database. The file also defines clear endpoints and manages API methods accordingly.

`/backend/Seeder.js`

It is used to fastly & efficiently insert and delete data to the database.