

CSCC01 eBeauty Project Documentation

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

July 2, 2022

Table of Contents

Frontend.....	4
data/db.js.....	4
src/actions/taskProviderAction.js.....	4
src/components/AddressForm.js.....	4
src/components/AddressPage.js	4
src/components/Animation.js.....	4
src/components/FindJob.js	4
src/components/Footer.js.....	4
src/components/Loader.js	4
src/components/Login.js.....	4
src/components/MapContainer.js.....	4
src/components/Message.js	5
src/components/Navbar.js.....	5
src/components/NotFound.js.....	5
src/components/Paginator.js	5
src/components/ProfileProvider.js.....	5
src/components/ProfileCard.js.....	5
src/components/ScrollToTop.js.....	5
src/components/SearchBox.js	5
src/components/SearchPage.js	5
src/components/ServiceDropdown.js.....	5
src/components/SignUp.js	6
src/components/SignUpProvider.js	6
src/components/SignUpProviderTwo.js.....	6
src/components/SignUpProviderThree.js	6
src/components/ViewProfileProvider.js	6
src/constants/taskproviderConstant.js	6
src/helper/Alerting.js.....	6
src/pages/HomePage.js	6
src/pages/ReviewPage.js	6
src/reducers/taskproviderReducer.js.....	6
src/App.js & src/index.js	7

src/store.js	7
Backend	7
/backend/config/db.js	7
/backend/controllers/customerController.js	7
/backend/controllers/postController.js	7
/backend/controllers/providerController.js	7
/backend/controllers/reviewController.js	7
/backend/controllers/taskProviderController.js	7
/backend/data/profiles.js	7
/backend/data/service.js	7
/backend/data/taskProviders.js	8
/backend/middleware/authMiddleware.js	8
/backend/middleware/errorMiddleware.js	8
/backend/middleware/upload.js	8
/backend/models/postModel.js	8
/backend/models/providerModel.js	8
/backend/models/reviewModel.js	8
/backend/models/serviceModel.js	8
/backend/models/taskProviderModel.js	8
/backend/routes/customerRoute.js	8
/backend/routes/imageRoute.js	8
/backend/routes/postRoute.js	9
/backend/routes/providerRoute.js	9
/backend/routes/reviewRoute.js	9
/backend/routes/taskproviderRoute.js	9
/backend/Server.js	9
/backend/Seeder.js	9

Frontend

data/db.js

It is the sample data in json format. being used inside seeder.js to efficiently load/delete content inside this file to the database.

src/actions/taskProviderAction.js

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

src/components/AddressForm.js

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

src/components/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/components/Animation.js

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

src/components/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/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/Login.js

This file allows existing users to login to their account.

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/NotFound.js

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

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/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/components/ProfileCard.js

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

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/SearchPage.js

This file is the component for a search result page, it should 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/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/components/SignUp.js

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

src/components/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/components/SignUpProviderTwo.js

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

src/components/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/components/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/constants/taskproviderConstant.js

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

src/helper/Alerting.js

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

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/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/reducers/taskproviderReducer.js

This file is used to request reducers for all task providers and a single task provider. It returns loading state status and taskprovider 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/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/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 a function to get a provider's info (this is only used for posts, in future there might be necessary).

/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/controllers/taskProviderController.js

It contains the functions to request task provider data from DB, two apis in this file, one to get all task provider data, the other one to get a single task provider details by its id.

/backend/data/profiles.js

sample data of profile schema

/backend/data/service.js

sample data of service schema

`/backend/data/taskProviders.js`

sample data of taskproviders schema

`/backend/middleware/authMiddleware.js`

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

`/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 could 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/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/models/taskProviderModel.js`

schema of taskProvider for mongoDB

`/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/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/routes/taskproviderRoute.js`

It defines the endpoints of task provider DB query activities. Routing the method to function in taskproviderRoute

`/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.