Objective of the assignment was to create a web app that is used to display recipes.

Hangry is a simple webapp used to search recipes, let's user add his own recipes.

Workflow of the website in Users PoV:

1. The user can login or sign up on the website.
2. Users can search for recipes by going to the search page.
3. The search results are displayed and the user can click on the recipe they like which takes them to the new page which shows the details of ingredients and procedure.
4. Users can later go to Add recipes to add new recipes which creates a personal cookbook for individual users in the dashboard.
5. User can find his own recipes in the dashboard and can click on them to get the details
6. Users can press edit to edit the ingredients. For now only ingredients can be updated.

Workflow of the website in Developer PoV:

1. User auth is handled by Firebase and Firestore making it easy for small applications for user management
2. In order to let the user search and get results of the recipes a 3rd party API called *spoonacular* is used. The api provides complete details of the recipes including many details such as type of cuisine, calories, summary etc. For this assignment it has been limited to the title, image, ingredients and procedure.
3. A form is provided for the user to add his recipes and this is stored in Firestore.
4. In the dashboard only those users who add their recipe can view and edit their recipes.
5. To keep the UI simple and robust Materialize is used to make the website look presentable without adding over the top CSS.
6. For the front end React + Redux is used while for backend Firebase is used. Well NodeJS could be used for backend I stuck to Firebase as it gave more time to concentrate on the React and redux part making it faster to build the website and not worry about the user auth.
7. All the routing within the website is done using react-router.
8. Route guarding is implemented on few routes making it accessible to only those who are logged in. If not they are routed to Sign up page

Different components used:

1. Auth : This contains the required components for login and signup.
2. Dashboard: This component takes care of the dashboard.
3. Landing-page: This component takes cares of the homepage
4. Layout: This contains the Navbar component, Nav links for different users who are logged in and signed out.
5. Recipe: This contains all the required components for user recipes which he adds
6. Search: This contains all the search apis and their related components.