Documentation for Zonion Application

Summary:-

Zonion app shows the active restaurant list and their corresponding details.

Admin can manage the restaurant by adding, updating, creating restaurant as well as there is feature where admin can activate and deactivate restaurant

Technologies for Zonion Application:-

* Node js
* Sails js
* MongoDB
* React Js

Backend Technologies used in Zonion Application:-

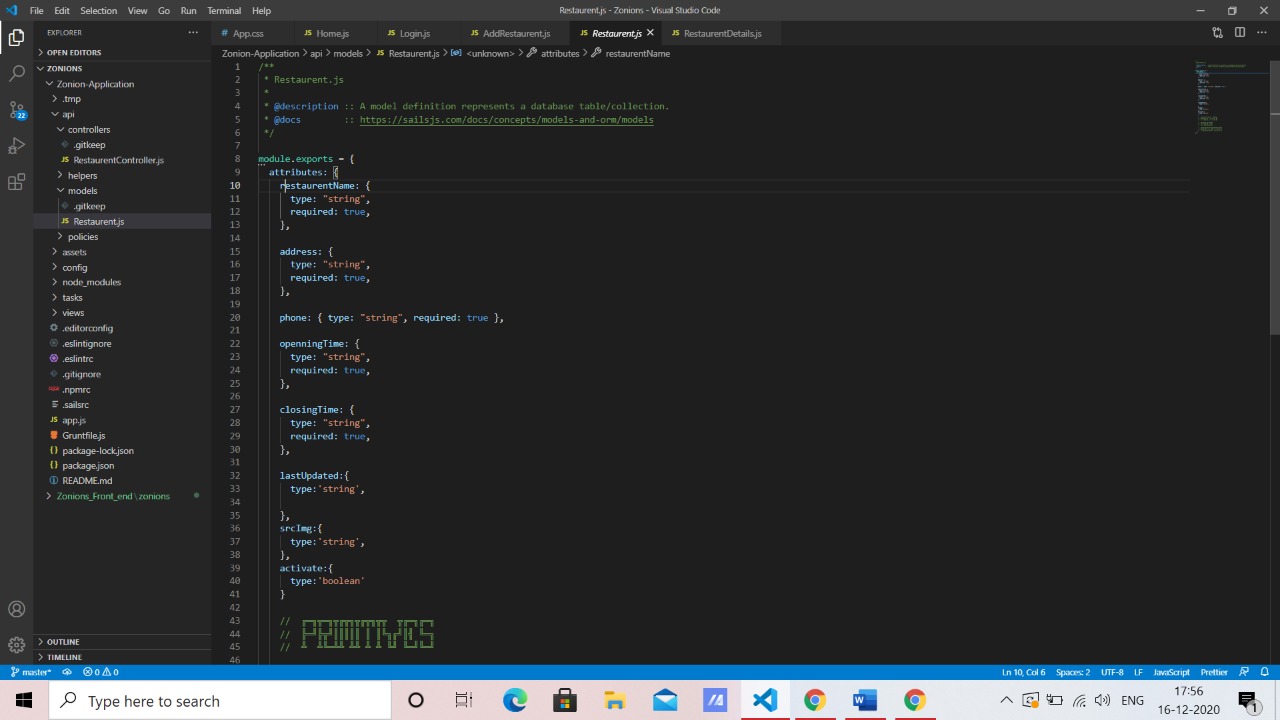
**Node.js** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), [back-end](https://en.wikipedia.org/wiki/Front_end_and_back_end), [JavaScript](https://en.wikipedia.org/wiki/JavaScript) [runtime environment](https://en.wikipedia.org/wiki/Runtime_environment) that executes JavaScript code outside a [web browser](https://en.wikipedia.org/wiki/Web_browser).It was used within this application for database connection and serving web-page.

Sails.js is designed to make it easy to build custom, enterprise-grade Node.js [web applications](https://en.wikipedia.org/wiki/Web_application) and [APIs](https://en.wikipedia.org/wiki/API).

MongoDb is NOSQL database used within thi application for storing managing the data.

Backend Structure for the App:-

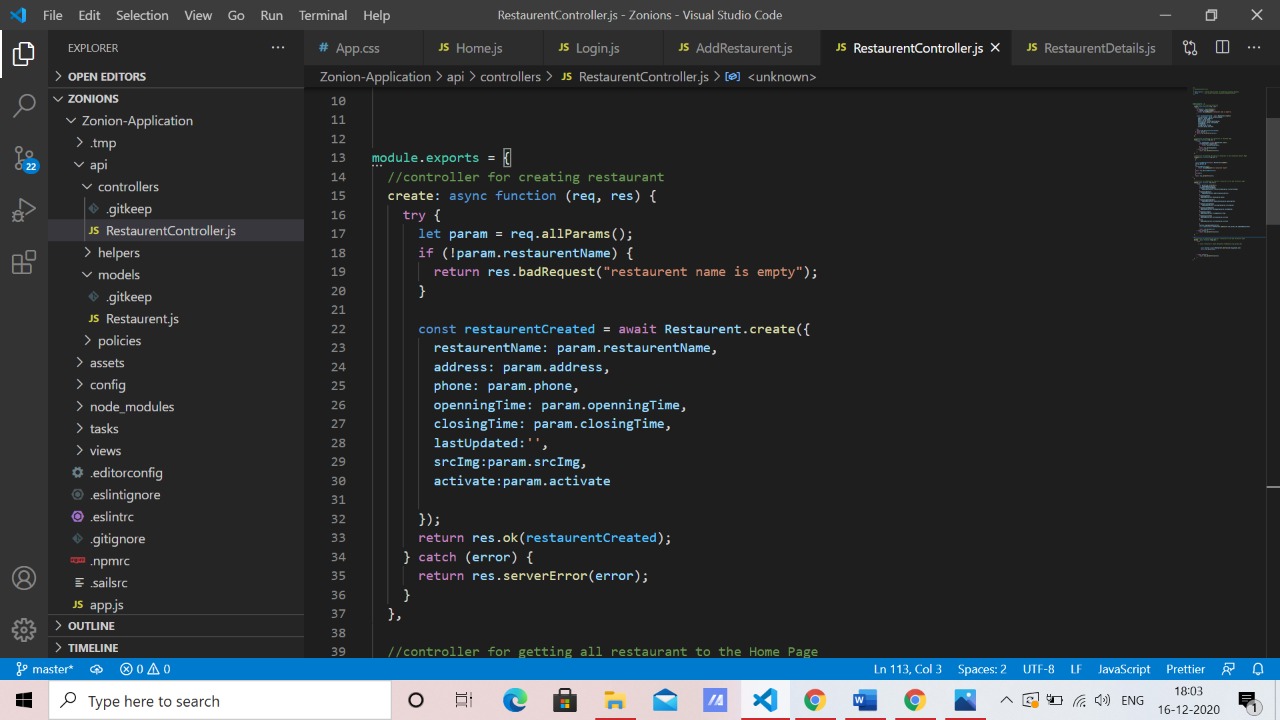
1.Model:-

Model for the database which shows the database what type of data is and some additional features within it 

2.Controller APIs

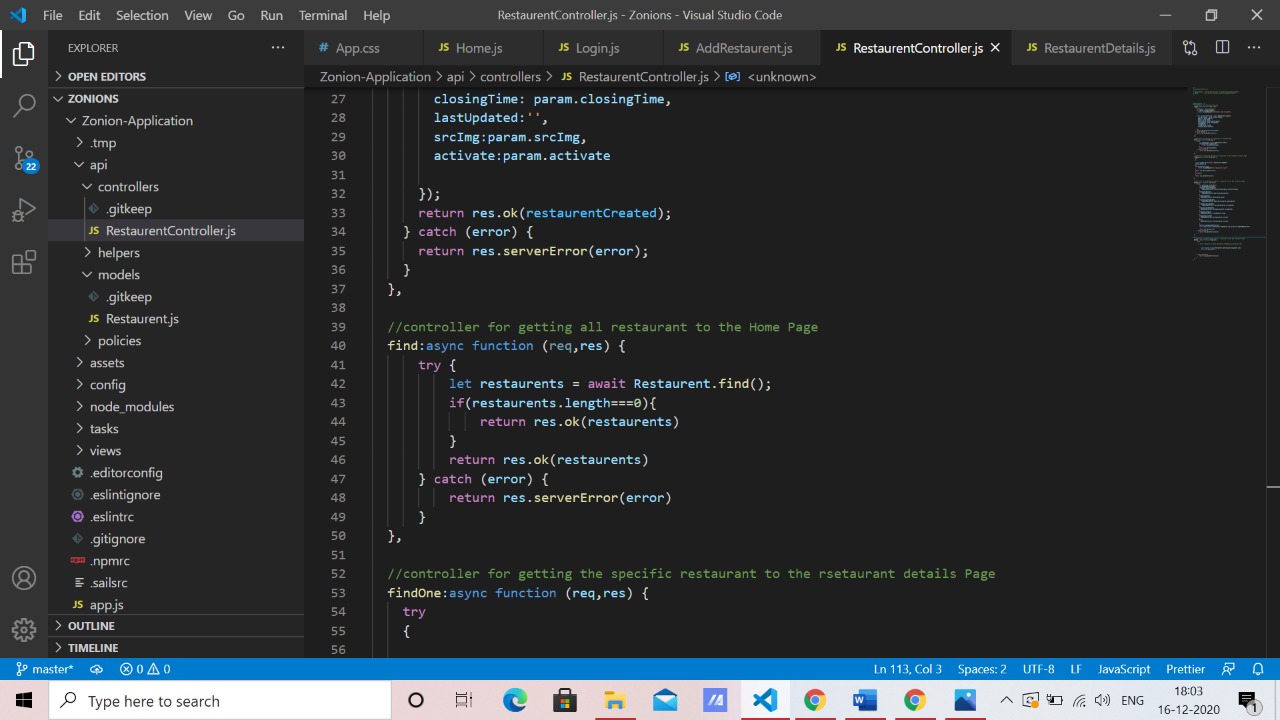
Controllers used for managing the restaurant details page that is creating , updating, finding all restaurant, finding a specific restaurant , deleting a specific restaurant.

1. Creating a restaurant



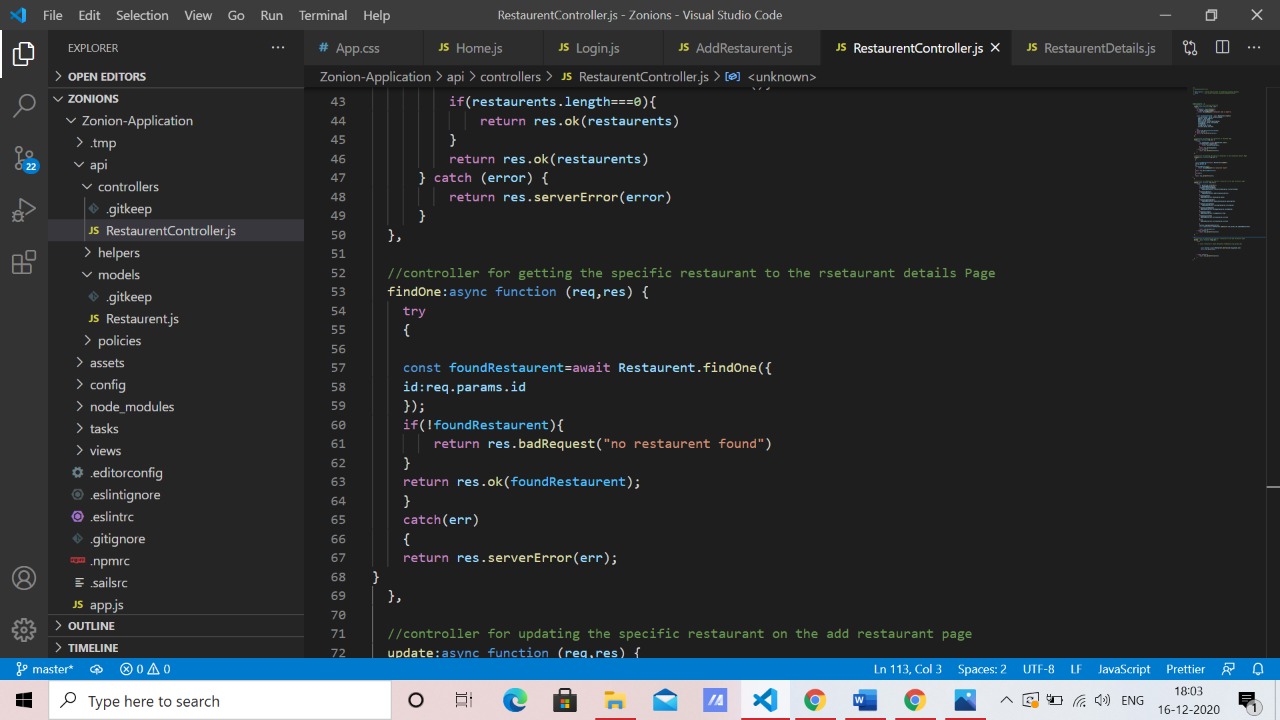
While creating restaurant it takes values from param and checks whether any required field is empty or not if there is any empty field then it throws response of bad request . After successful execution it shows restaurant created on the console.

1. Finding all the restaurant



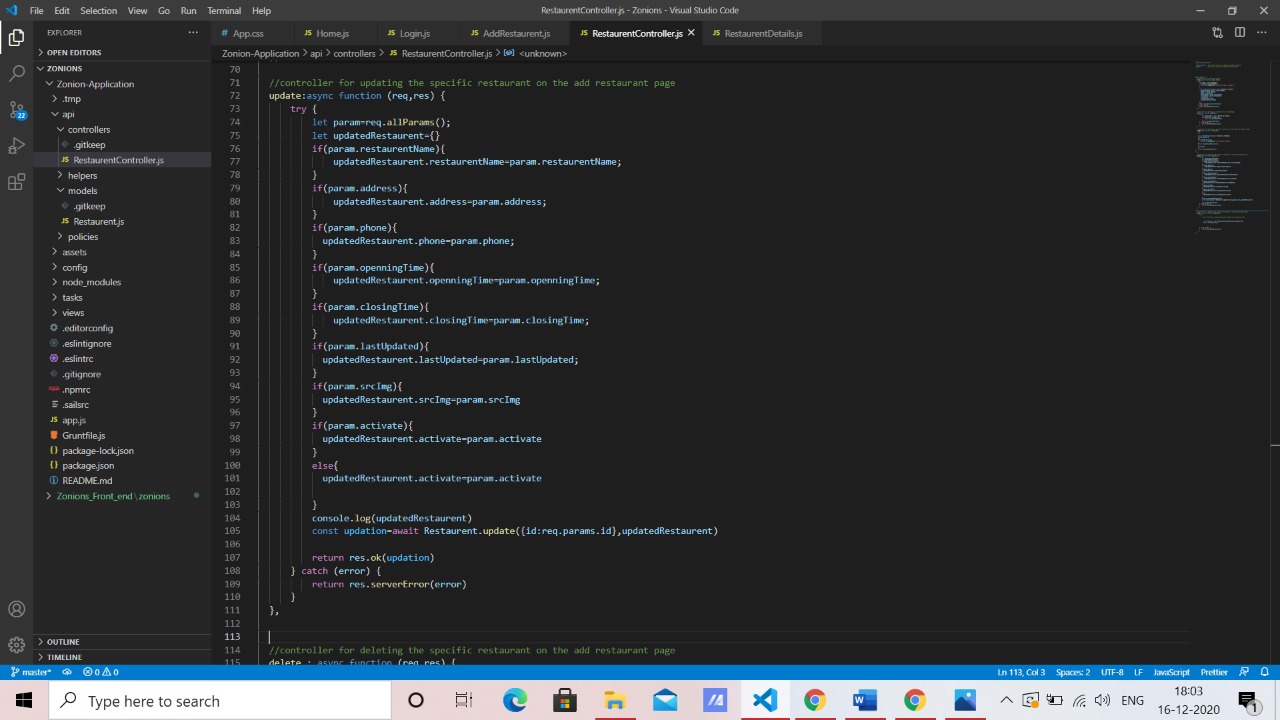
Finds all the restaurant from the database.

1. Finding the specific restaurant



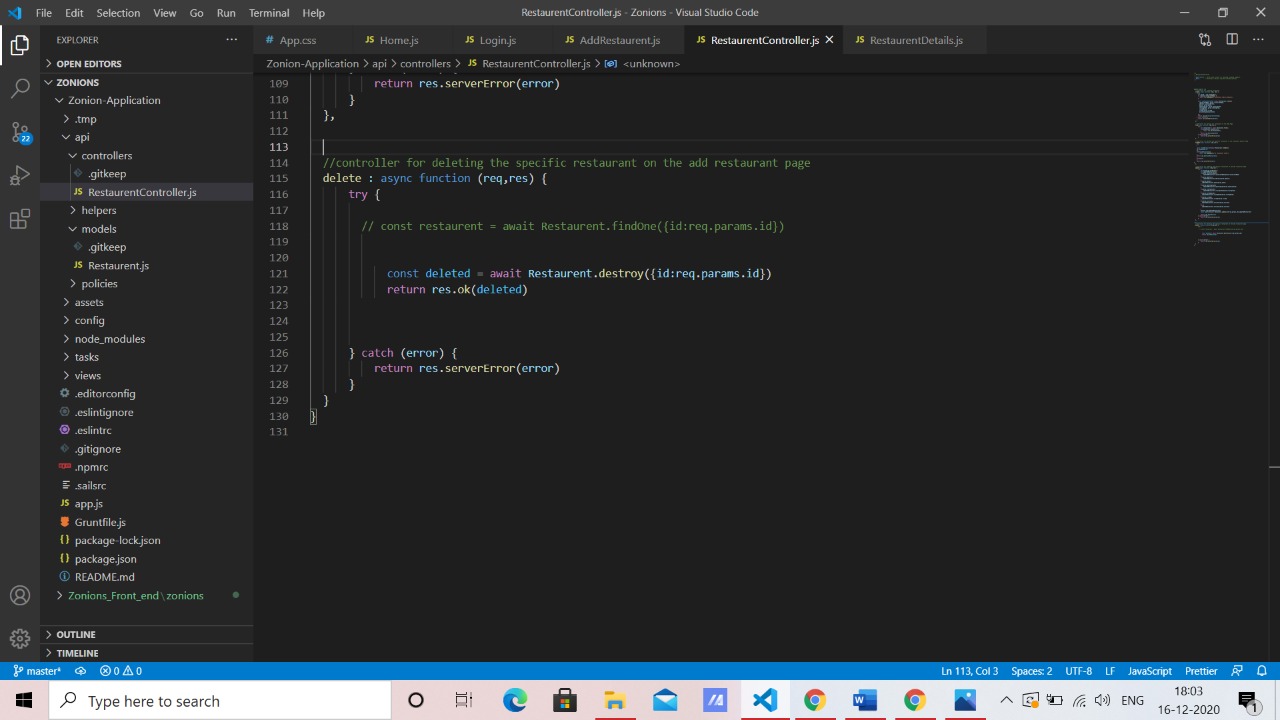
Finds specific restaurant from its \_Id where it checks every document with the specified id if match found then returns the result else response “no restaurant found”

1. Updating the specific restaurant



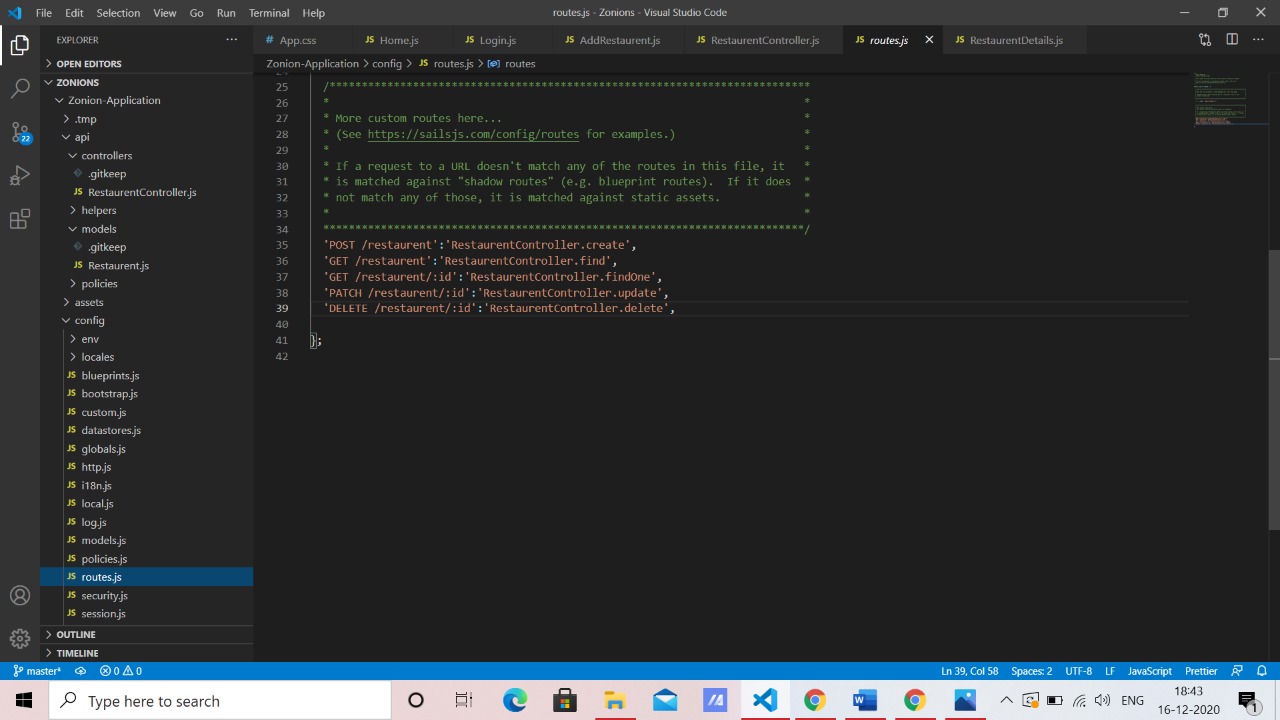
Updates the specific restaurant by matching its \_id with all documents and if match is found it checks for updates within param and if found then it updates it within existing mongo document

1. Deleting the specific restaurant



Deletes the specific mongo document by checking the \_id which is passed through param.

3. Routes:

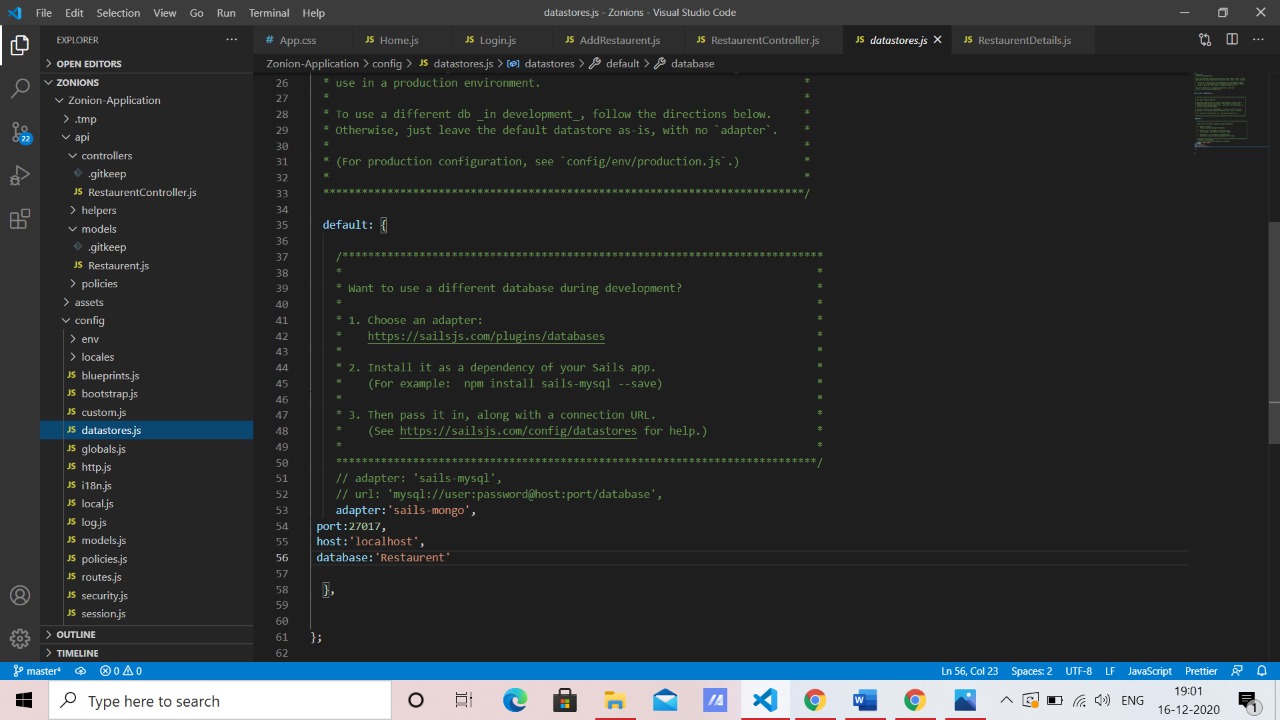


To fetch data and manipulate data controller is required and to access those controllers paths or routes are used .

Within the zonion app ,the data fetching and manipulation is done through third party library Axios and within axios pre-defined methods , we pass this routes so the app get to know which route has which controller .

It’s one of the most crucial thing within app because if there is typo error then axios will not able to find the controller and app will get crashed

4.Database connection

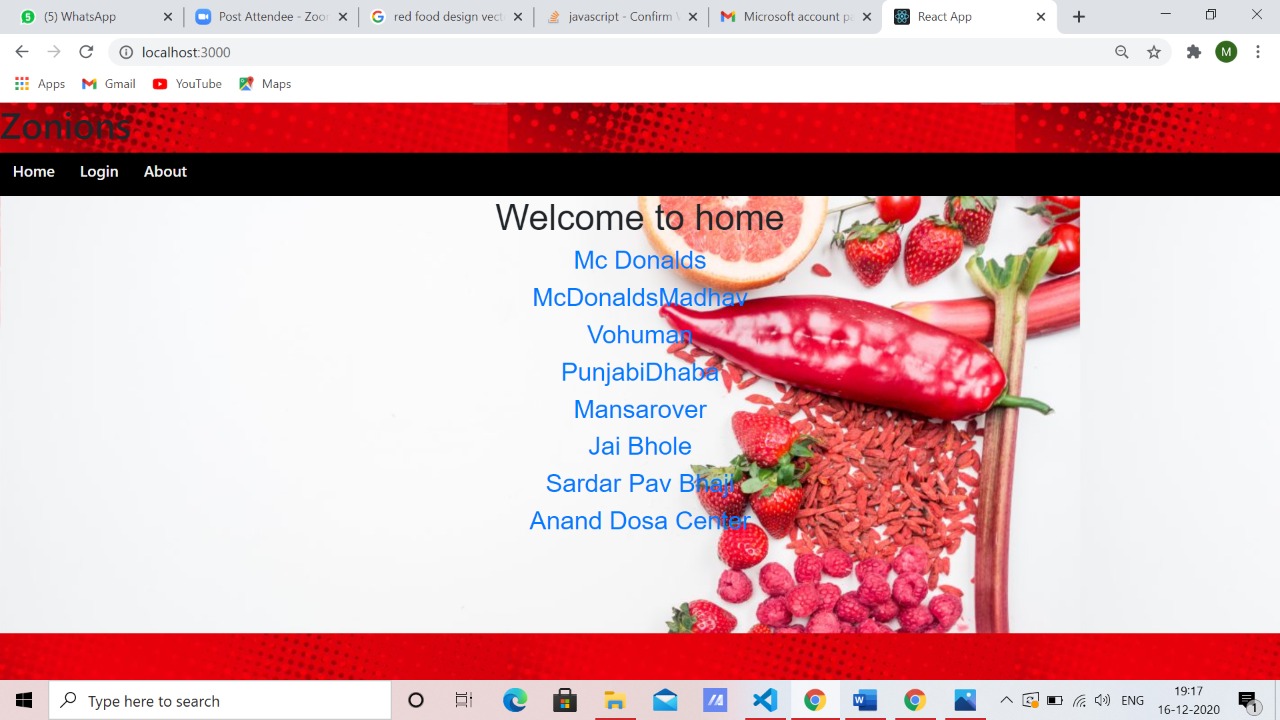


For the database connection we have to configure few things suchs as get dependencies ‘sails-mongo’ for adapter.

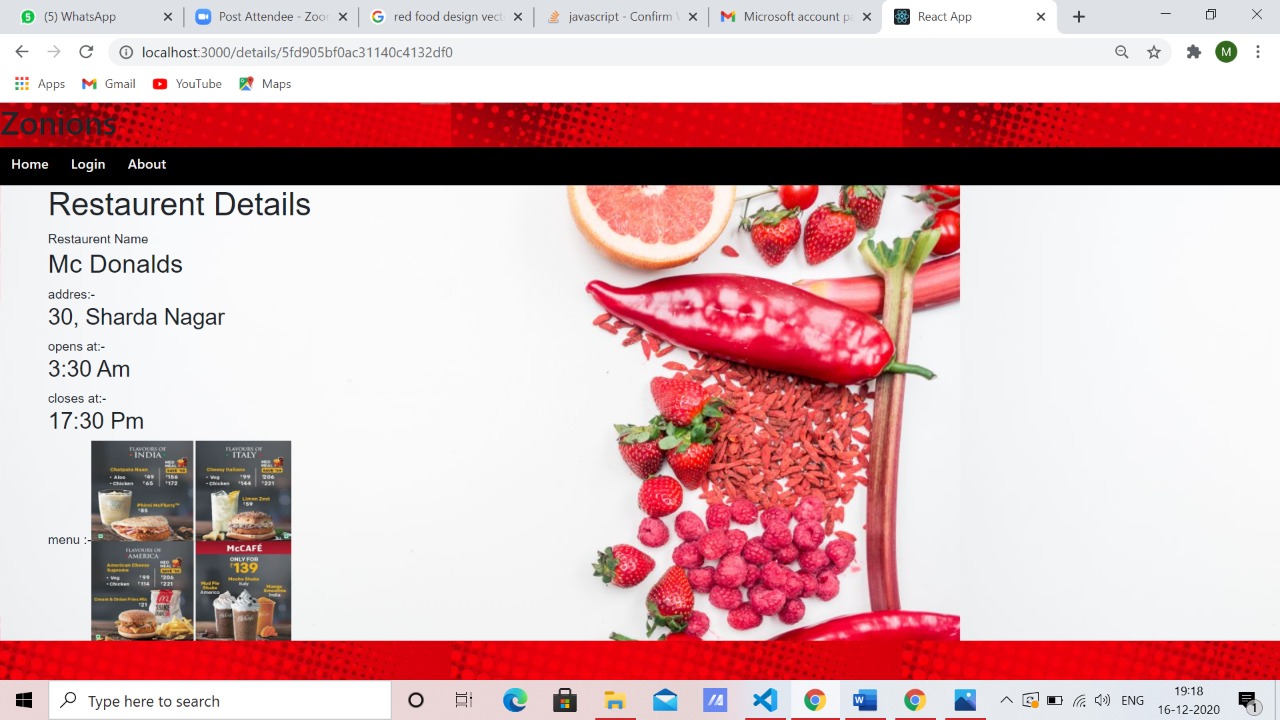
Front-End Structure For the Zonion

For the view of the app we have used React js

1. Components and its working

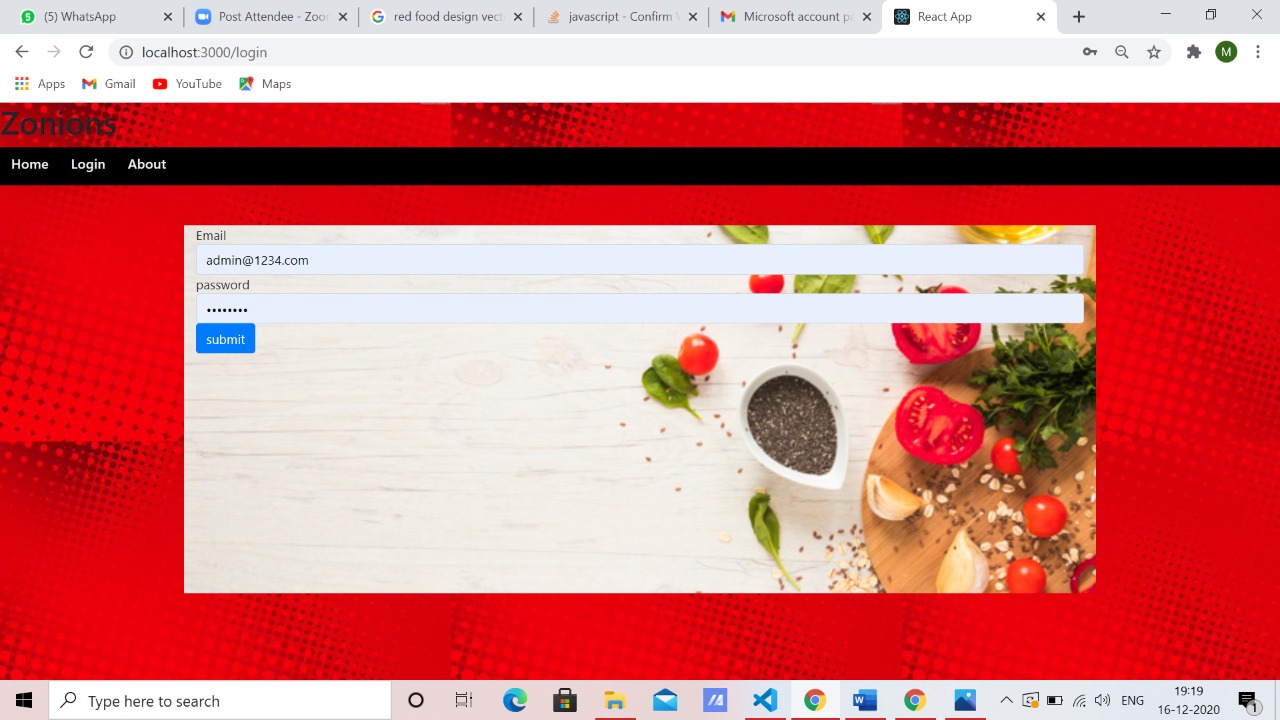


Home page where all the listed active restaurants are shown after clicking on one of the restaurant it will go restaurant details page where the details of that restaurant is shown



As shown in the above picture

On clicking the login tab it goes to Admin login page where it verifies the admin. In this app I have hard coded the admin but we can make separate Admins by maintaining its model within sails and providing separate Controller APIs and routes.



After admin login verification is done, it opens Add restaurant page where admin can add ,edit ,delete restaurant