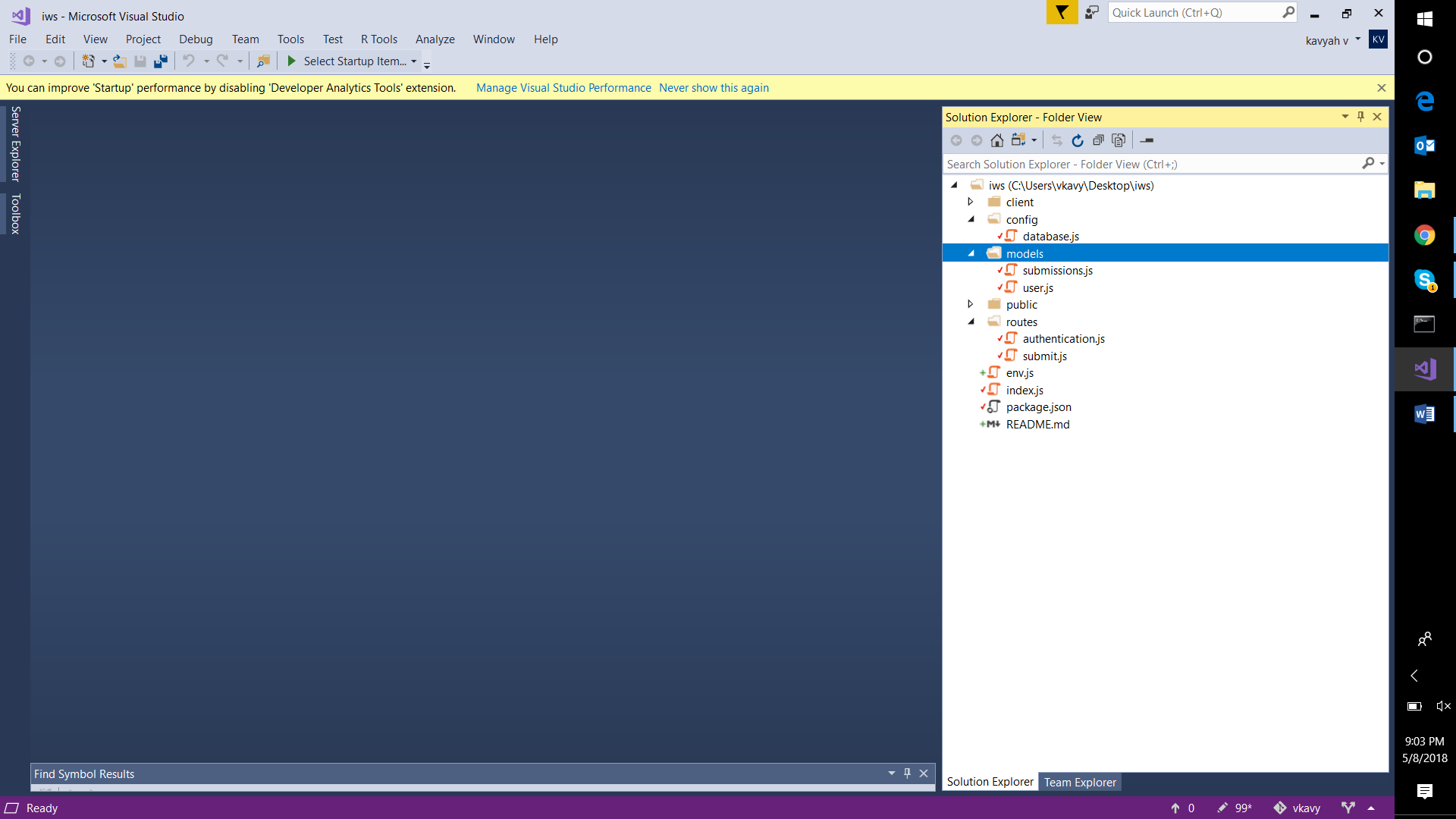
**Submission Portal**

**Report**



Server Side:

**Config Folder:**

It contains the database.js file that contains the url to the database and the database name along with the crypto code that encrypts the password.

**Models:**

The models contain the user schema in the **user.js** and the submission database schema **submission.js**. Validations for the server storage are also written in these files.

**Routes:**

**Authentication.js** contains routes and the Rest Api for the registration page. The post routes to create a new user , the get route to check if email exists in the database and the post route for login . It contains the validations for storing creating the new users and checking the login credentials in the route.

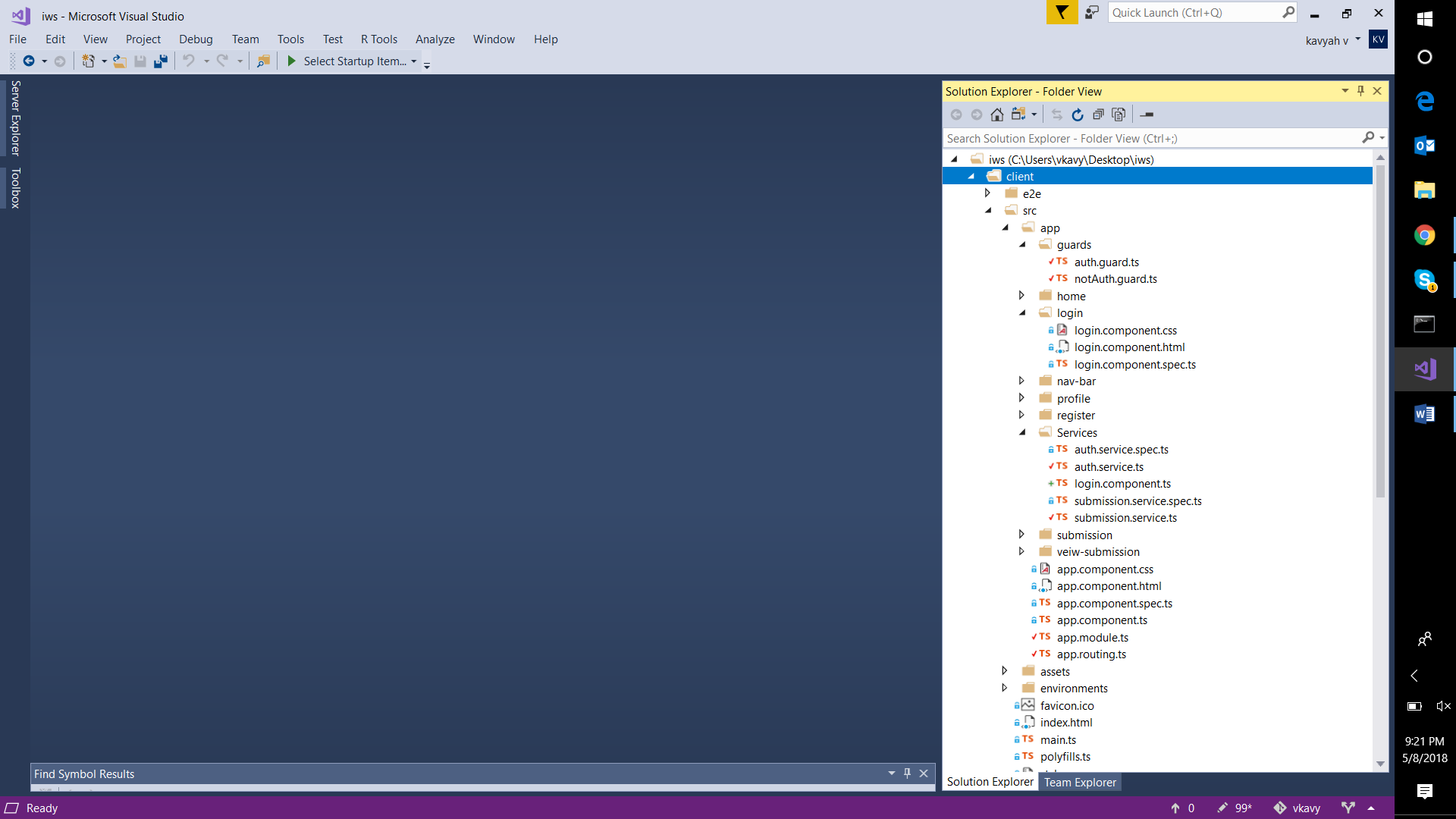
It contains the middleware which provides authentication to the routes and allows restricted access.

**Submit.js** contains the routes to create a new submission in the database using a POST route along with the validations for the submission form . It also contains a get route which allows access to display all the submissions.

**Index.js**

**I**t contains all the required routes and elements for the project . It establishes the connection of Express with Mongoose and defines a port on which the server can run . It provides the link to a static directory called **Public Folder** which is created when the project is built .

**The Front -End :Client Folder.**



**The Components:**

Following are the components of angular 2:

Home

Login

Profile

Nav-Bar – Bootstrap Navigation Bar

Register

Submission

View-Submission

Each of the component has 4 files . Work was done on the **html** file for the front-end view ,while the .**css** files are used to style the page using CSS. The **componentname.component.ts** files contain the validations and functionalities of the elements in the that are created in the html files.

**Services:**

The **auth.service.ts** file provides authentication services to the login and register component allowing only authorized users to access them. It also contains the code to create a Jwt token and hold it through the session.

The **submission.service.ts** file contains authentication for the submission and view submission components and subscribes to the **auth.service.ts** to provide the JWT token access to the submission database.

**Guards:**

The **auth.guard.ts** and **notAuth.gaurd.ts** are used to invoke authentication privileges on selected pages .

**app.module.ts**

It contains the collection of all the components and services used . It also contains the body of the NGModule.

**app.routing.ts**

It contains the paths to the pages and their authentication requirements. It is similar to the app.module.ts file but contains additional paths.