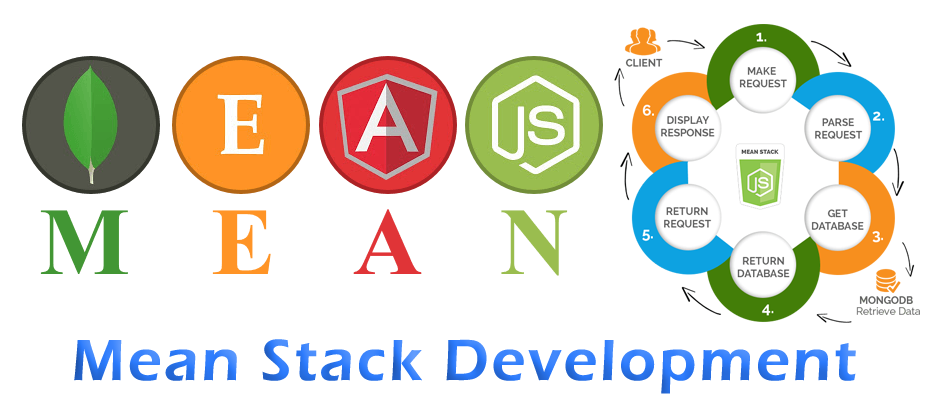


ASE\_Lab7 Report

Implement Customer CRUD using MEAN Stack





March 16, 2019

Documented by:

Dharani Muli (Class ID : 18),

Chakra Pavan Kumar (Class ID: 13)

Introduction

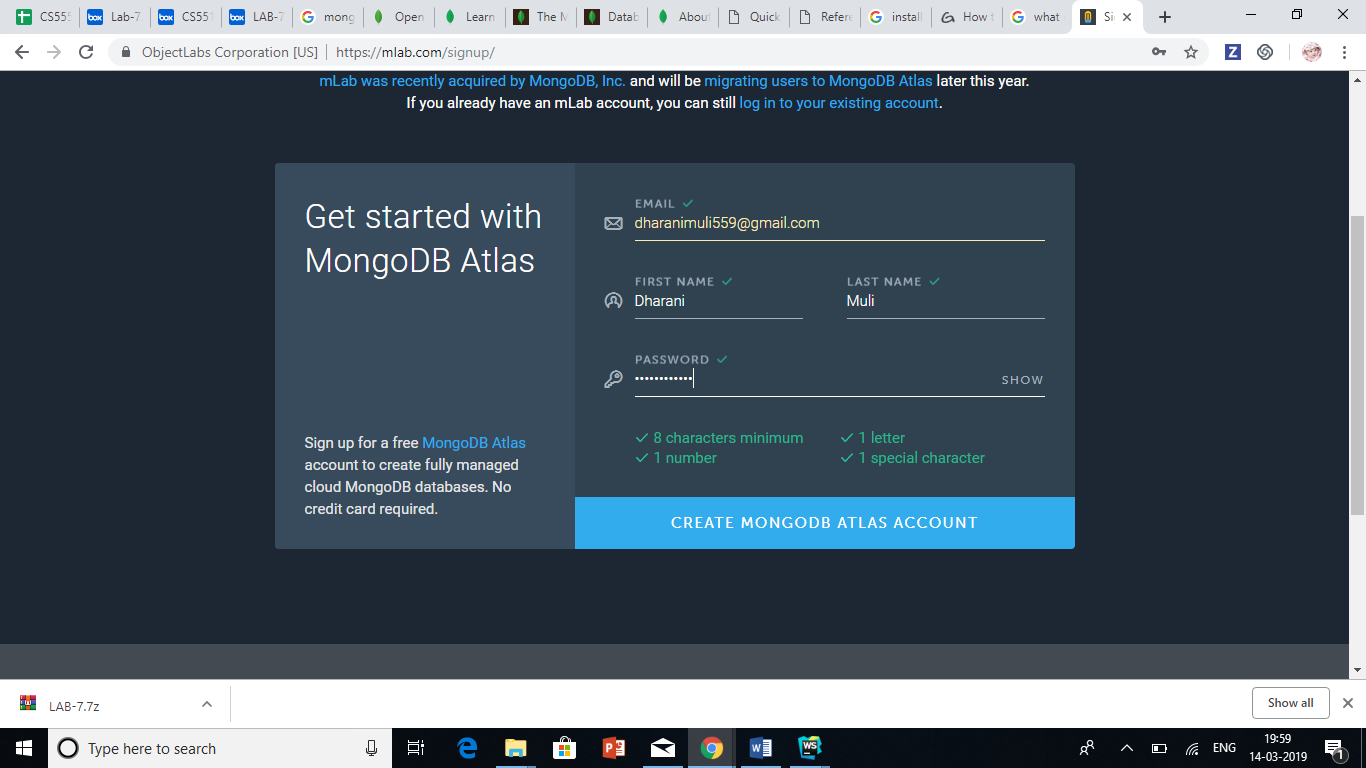
Objectives

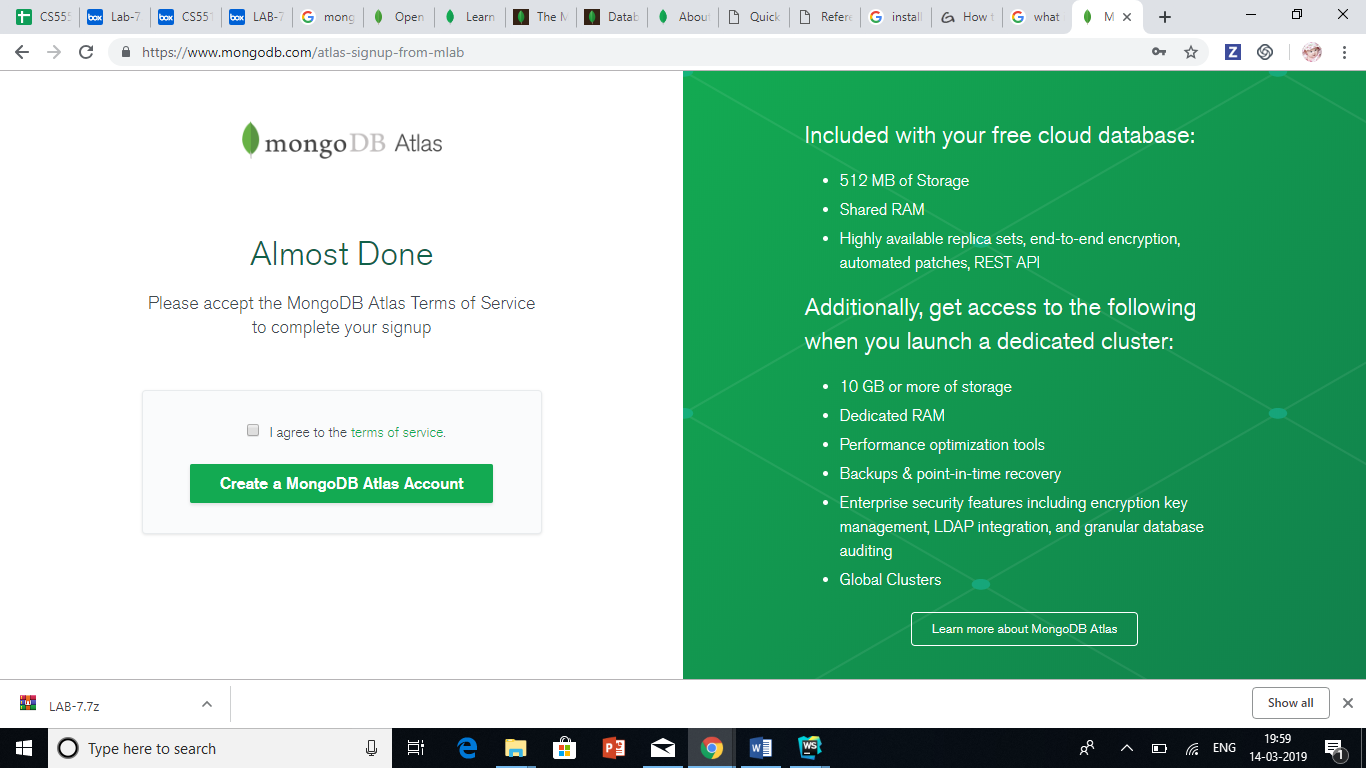
Develop customer CRUD API using MEAN Stack

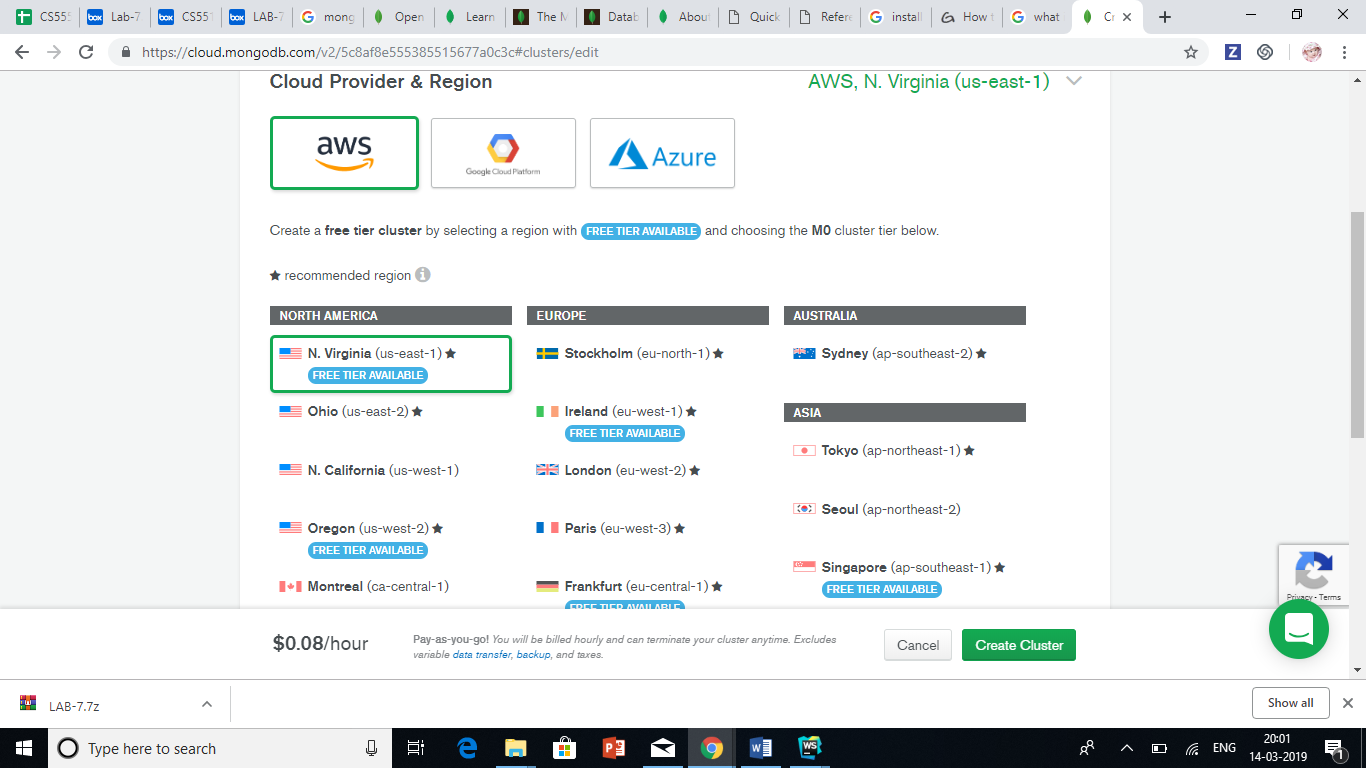
Design/Implementation

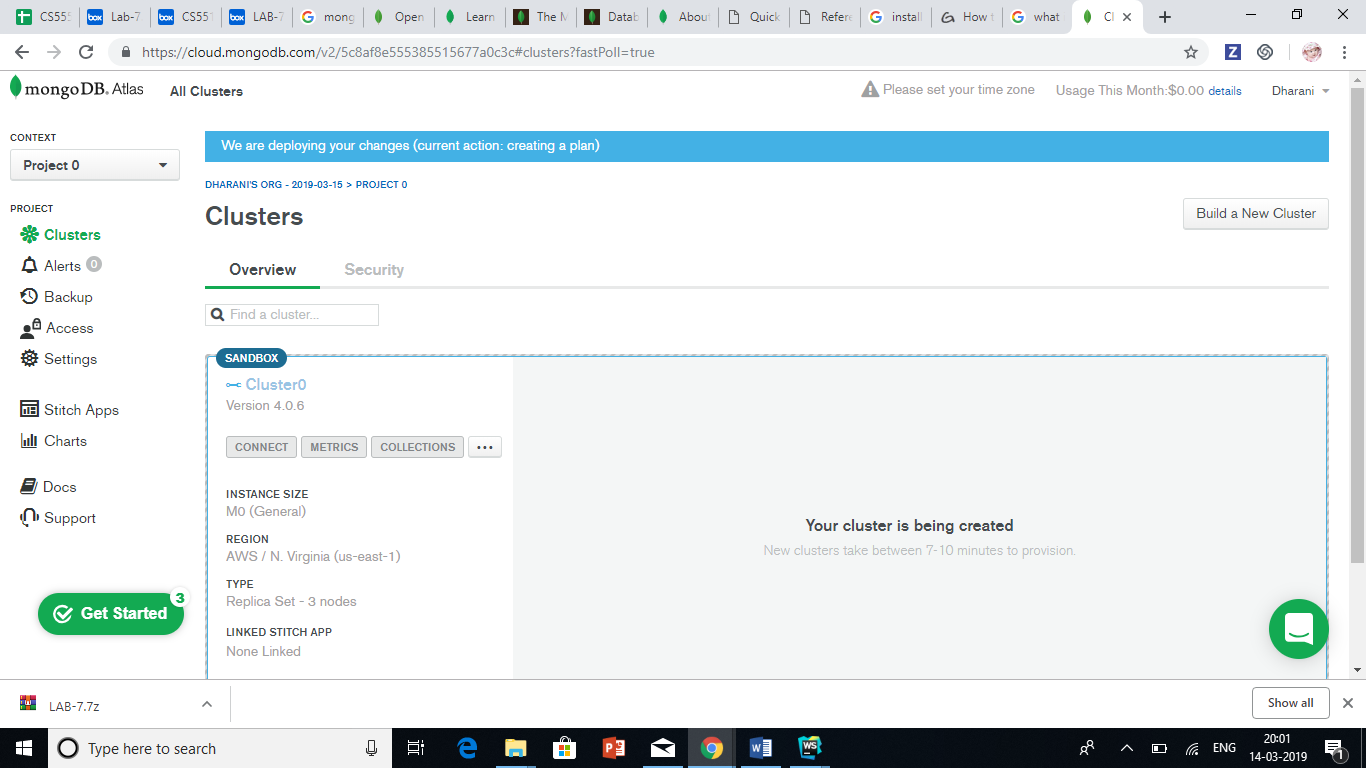
We have followed below steps to successfully complete this lab assignment:

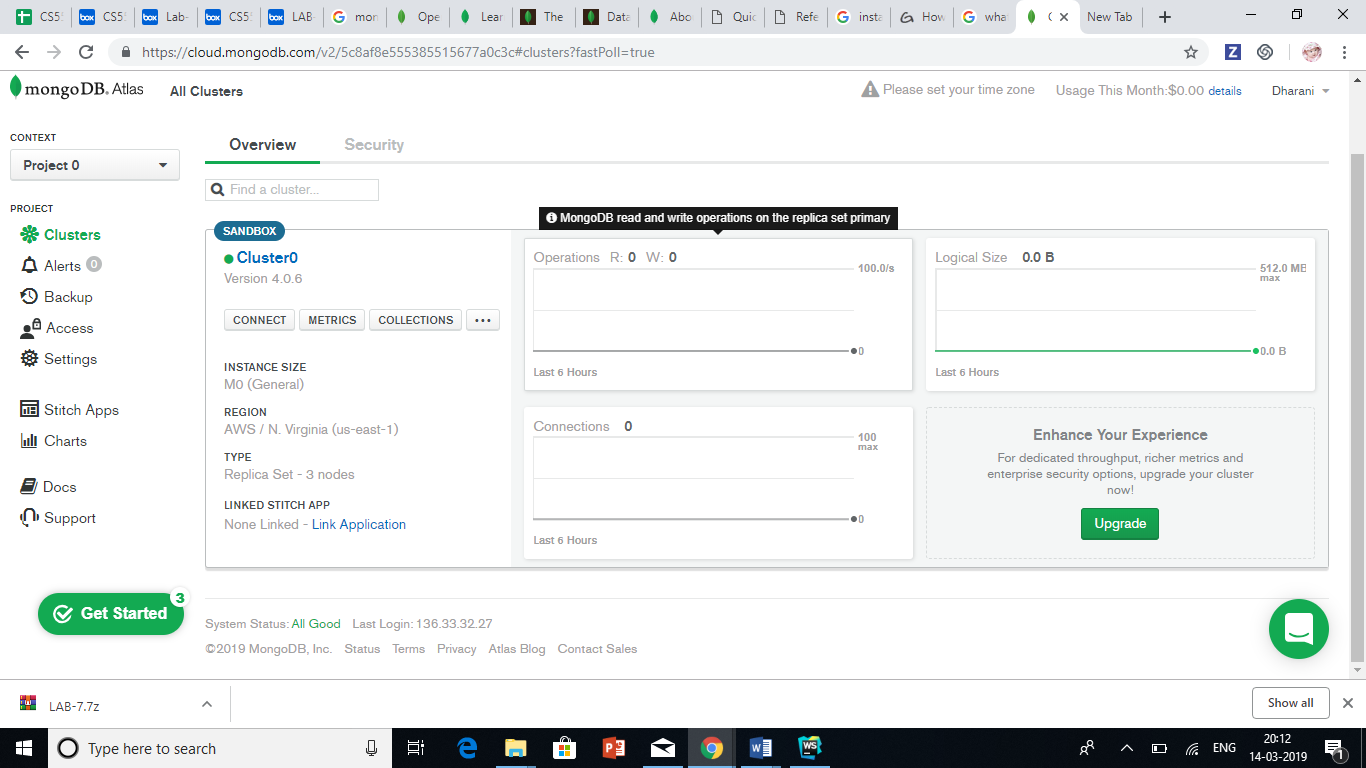
**Step -1:** We make sure all the pre-requisites are ready before start of the project and below are the technologies/languages used:

1. WebStorm IDE
2. Node
3. Npm
4. Expressjs
5. Angular
6. MongoDB: We should create Mongo Account and Create cluster. For that follow below Process:
   1. Create Account at <https://cloud.mongodb.com>

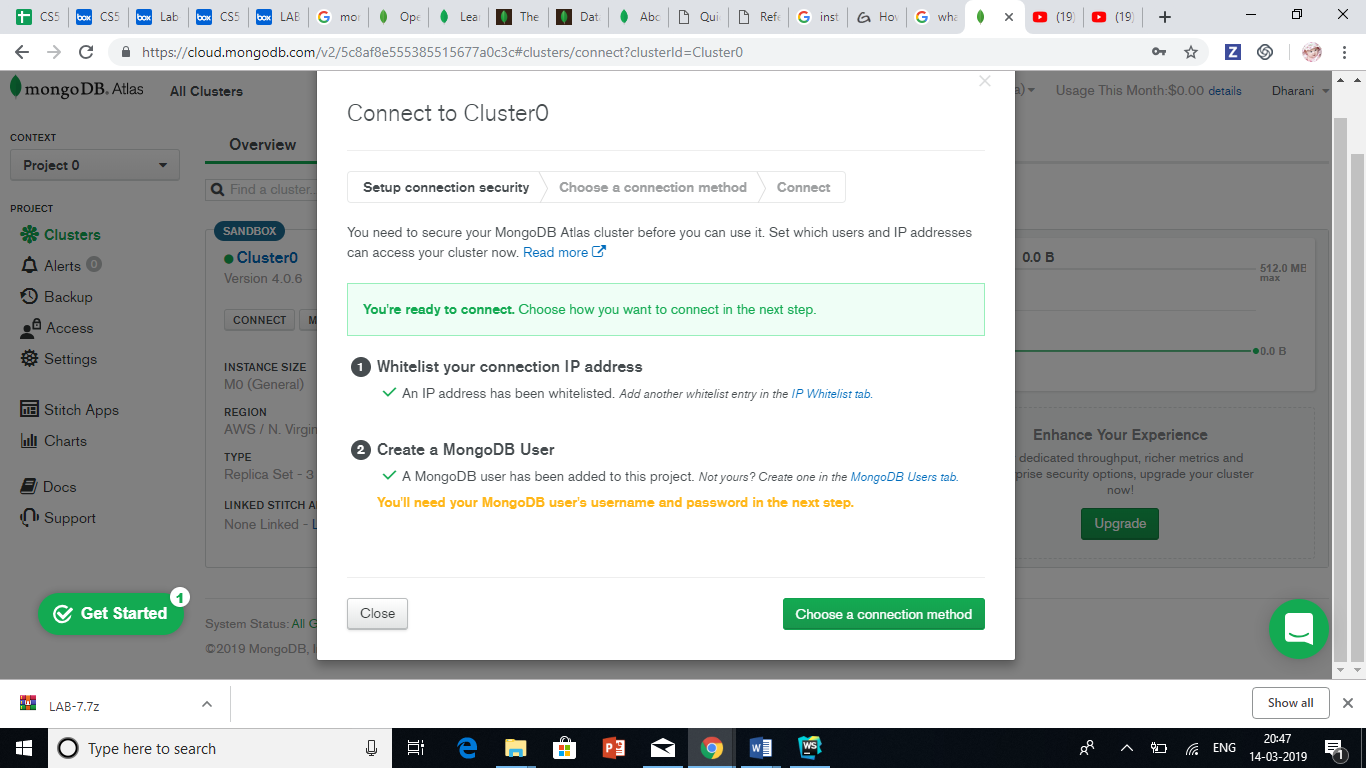




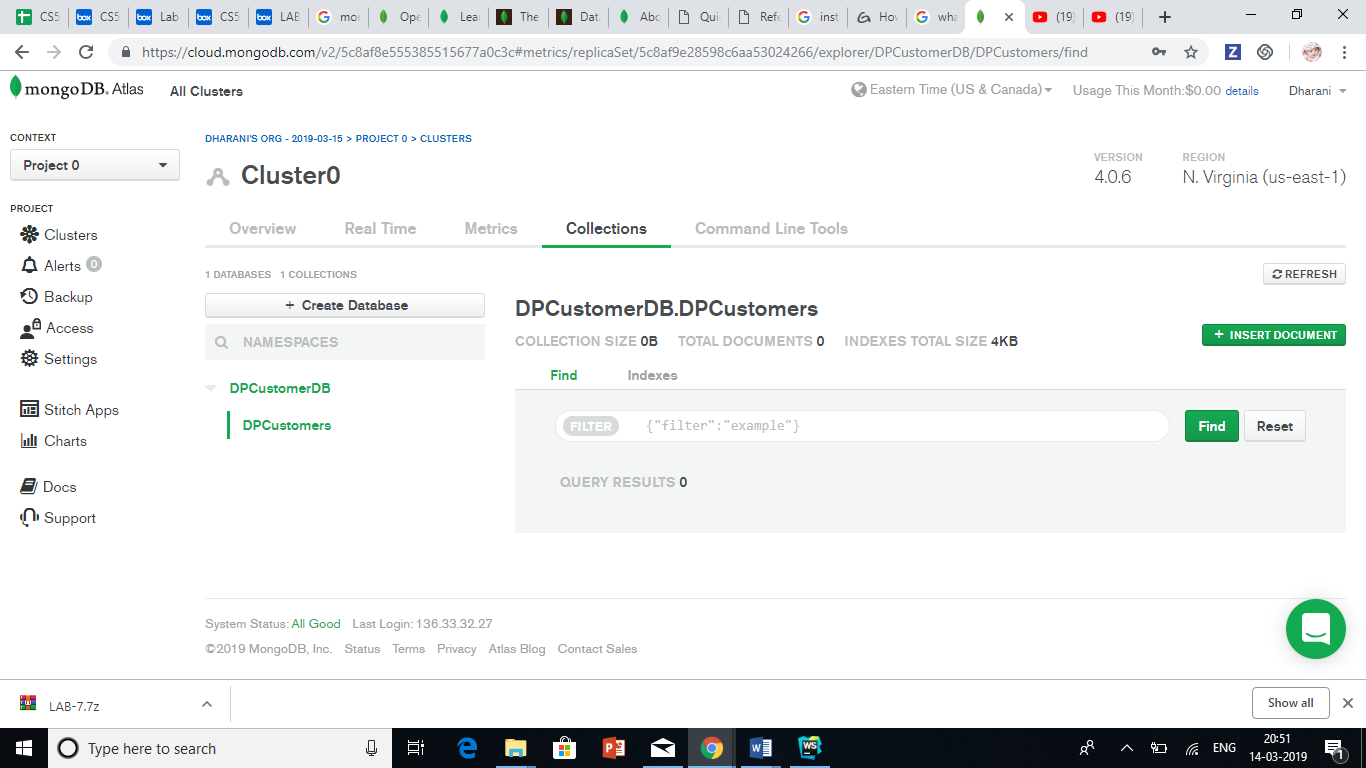




* 1. We need to set up connection security by Create User Name and Password (remember those credentials as they will be useful in future). Moreover, select “Allow from anywhere” which will allow us to connect from anywhere by providing username and password. Then you setup is done



* 1. Choose connection method “Connect your Application” as we are planning to build an application and then close the window. In this page you can see connection string which will be useful to code and connect to MongoDB.
  2. Click on “Collections” and Provide DB name and name of collection.

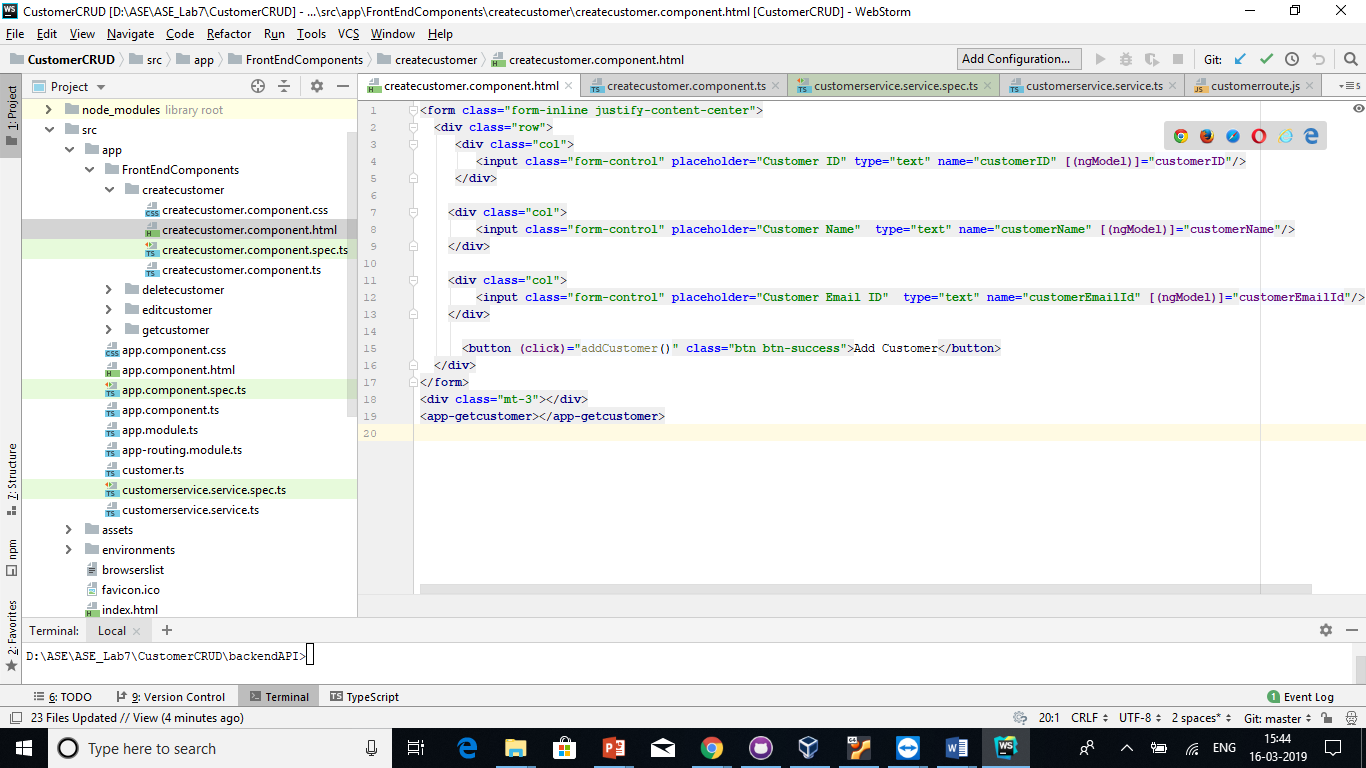


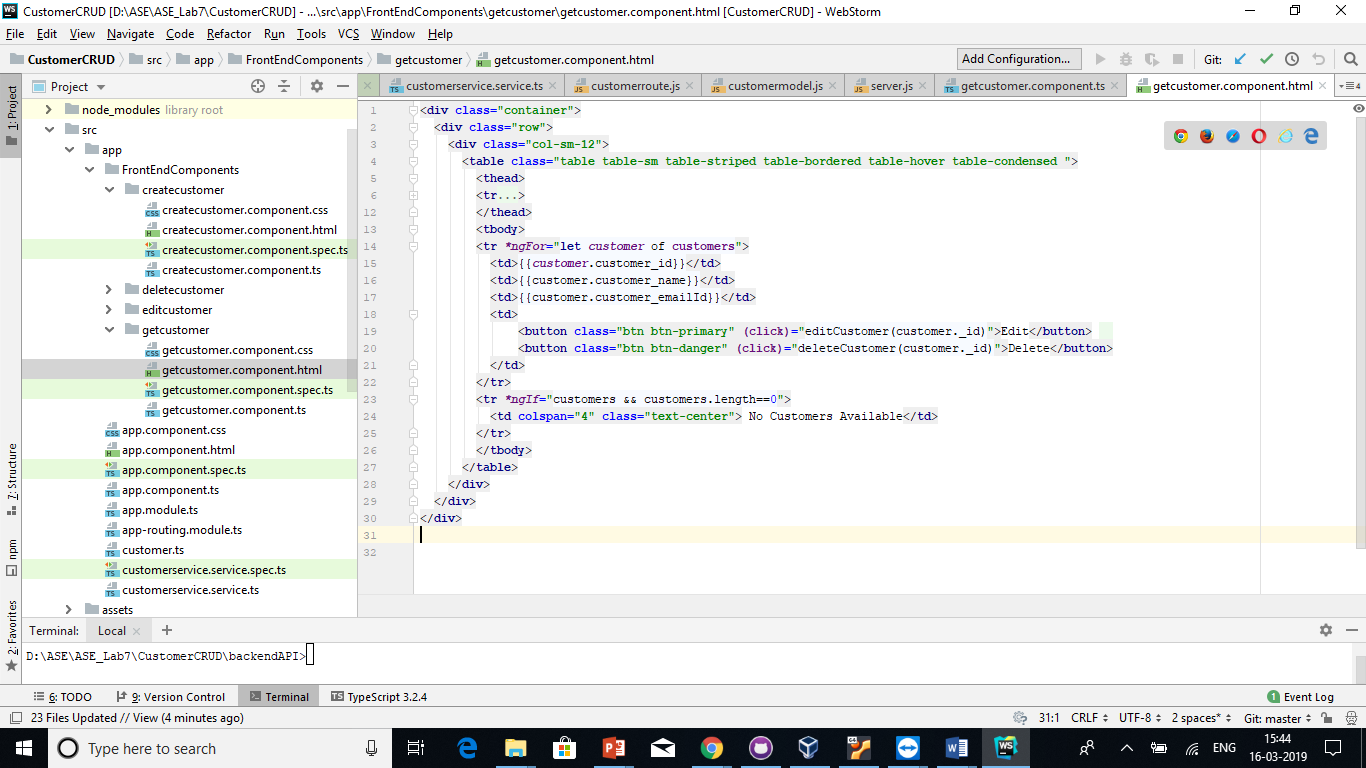
**Step-2:** As we are using Angular, we should create project using command “ ng new DPCustomerProject” and Using this command I have created components :

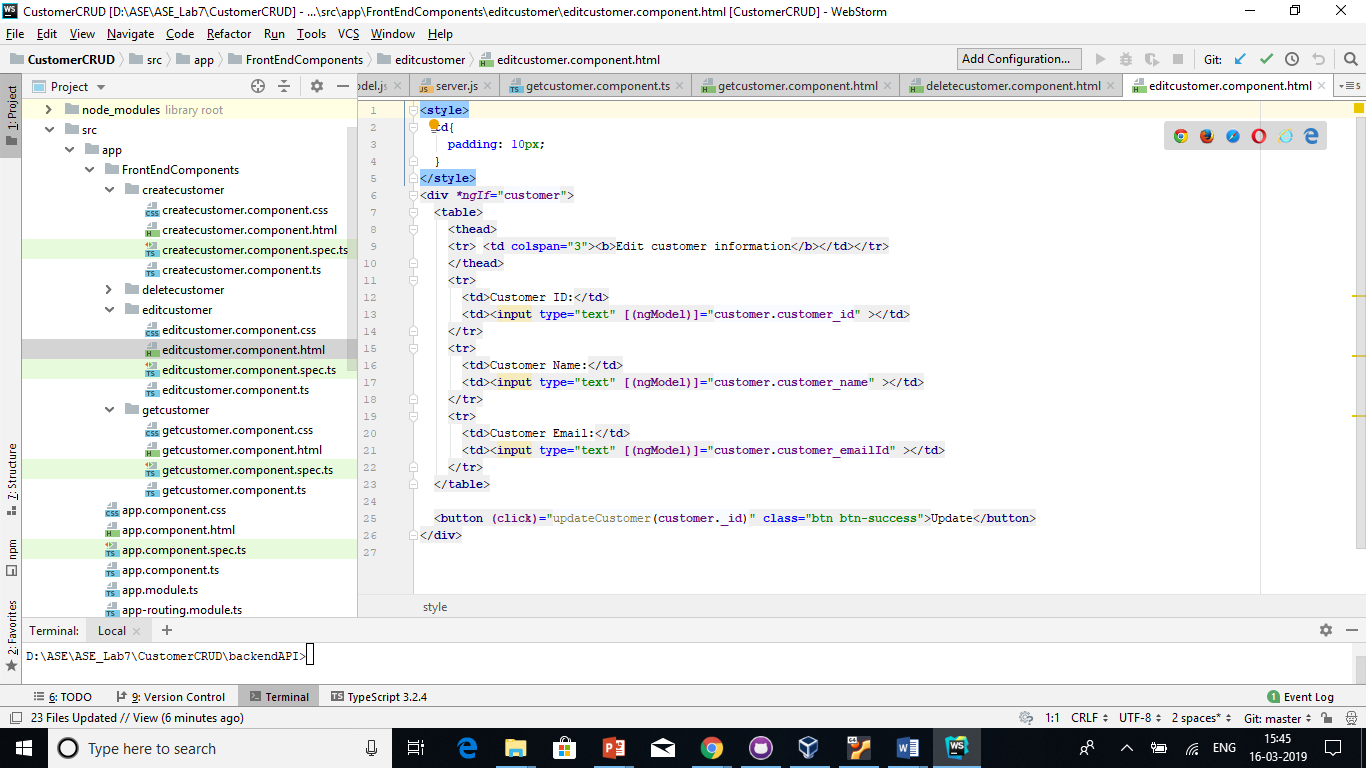
ng g c components/customerlist

**Step-3:** We have done the required configurations for components in the app.routing.module.ts file

**Step-4:** We have written below html code for Customer list, Adding Customer, Editing and Deleting Customer:

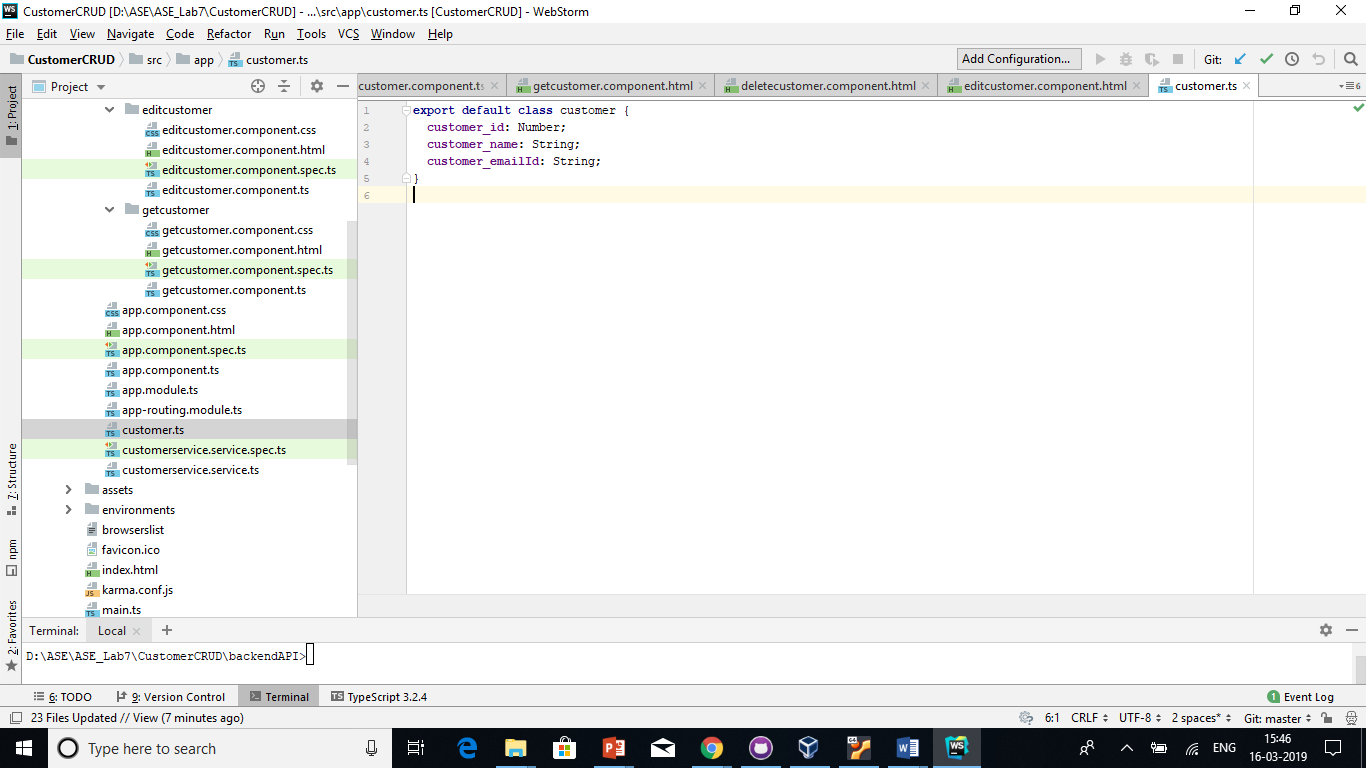
****

****

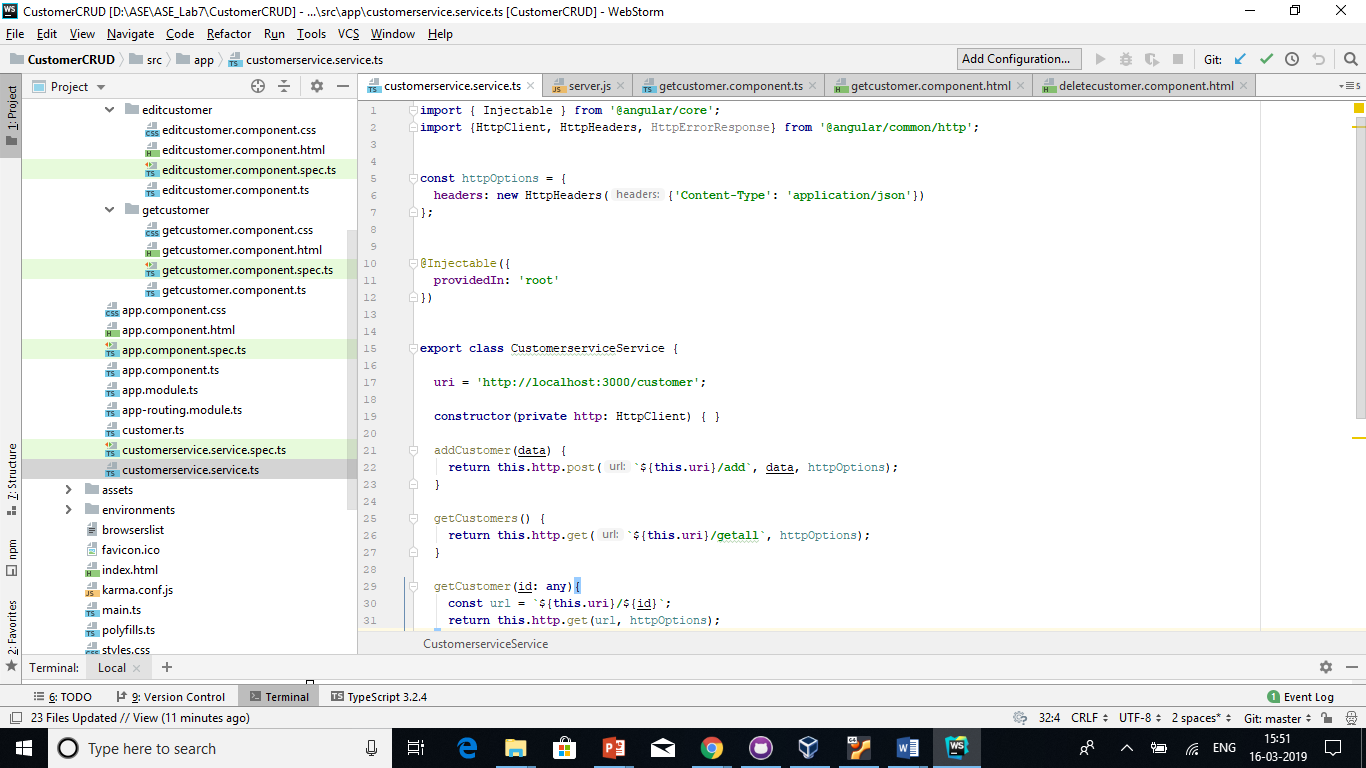
****

**Step-5:** Now we did the configuration of HTMLClientModule by importing the HTTPClientModule inside the app.module.ts file.

**Step-6:** Inside the src>>app folder we have created a file called customer**.**tsand add the following code to create a model at the front end

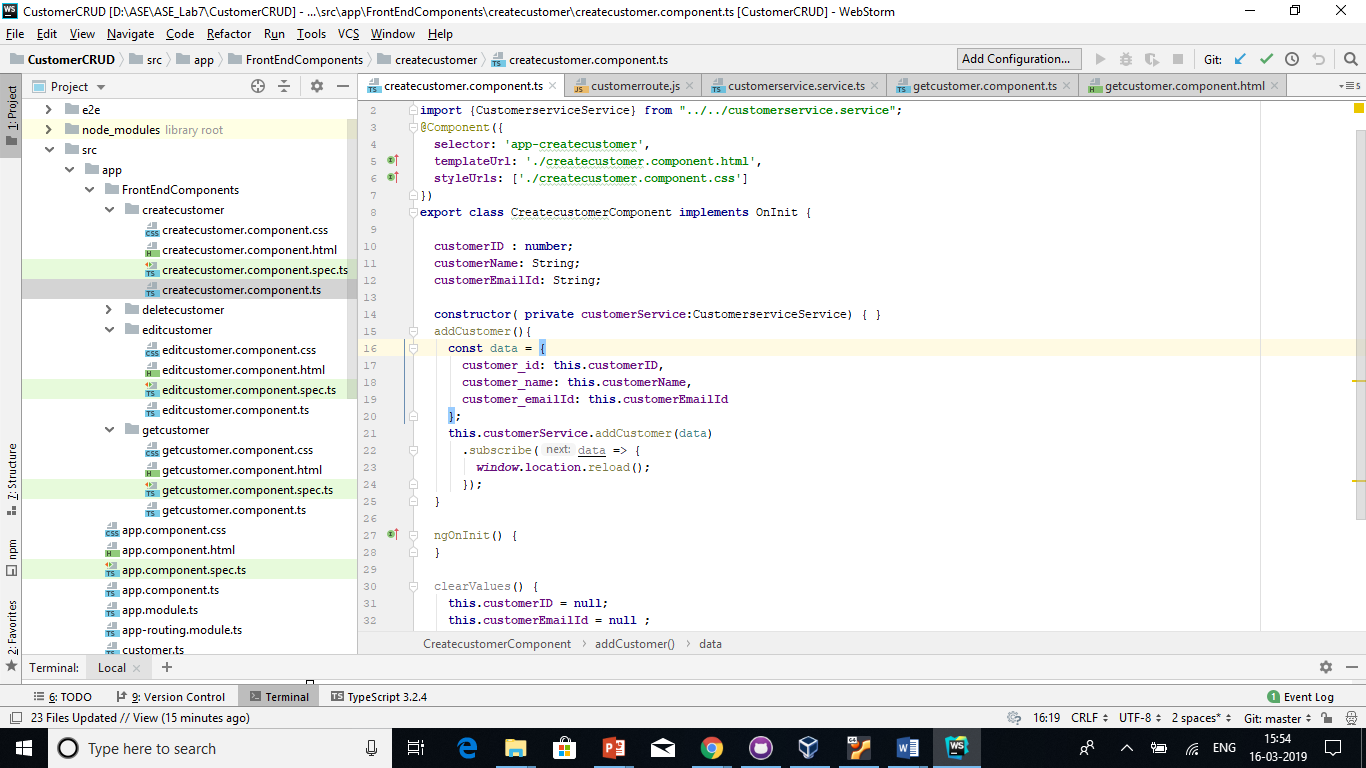


**Step-7:** Create angular service file by using the command “ng g service customerservice” and then imported this service file into app.module.ts. In this service we have written code that will send HTTP post request with the data to the Node.js server and save the data into MongoDB database.



We have defined out backend API URI but we haven’t created any backend yet in the initial stage but later we have done that.

**Step-8:** Create addCustomer function inside the createcustomer.component.ts file

****

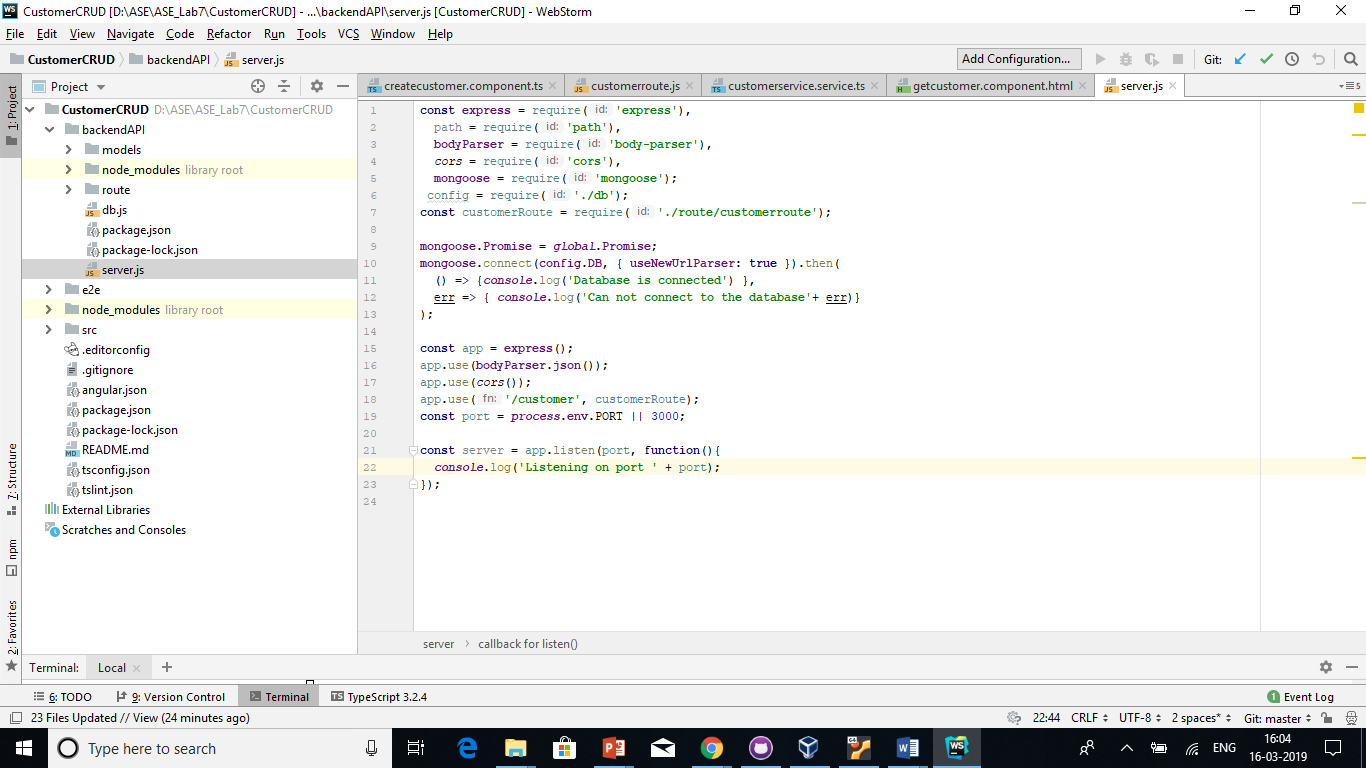
Here, we have defined the function and also imported the **customerservice.service.ts**file. Instantiate the object inside the.

**Step-9:** Inside the angular root folder, create **backendAPI** folder and go inside that folder. Remember, it will be a completely separate project from Angular. So its **node\_modules** are different from an **Angular**.

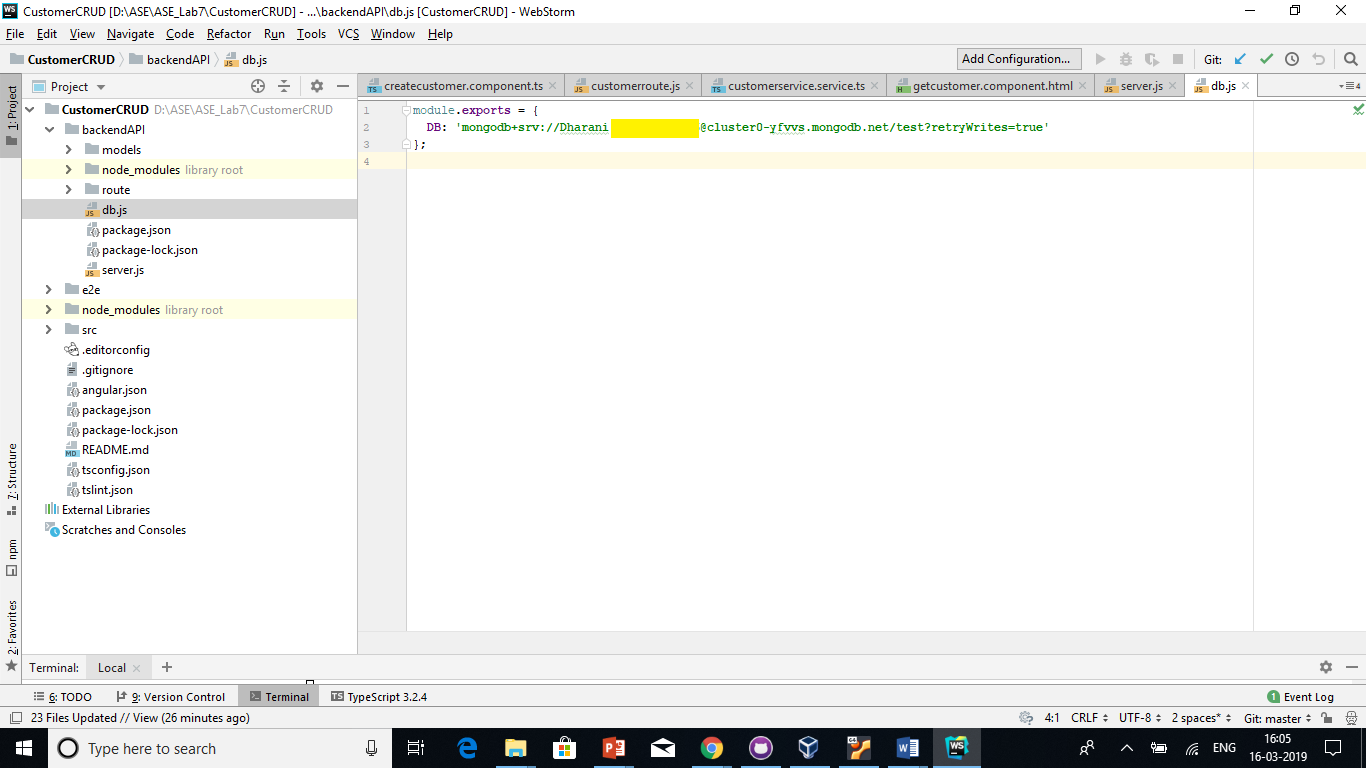
Open the terminal inside the **backendAPI**folder and type the following command : npm init

which will create package.json file and also installed the nodemodeules by using the command: npm install.

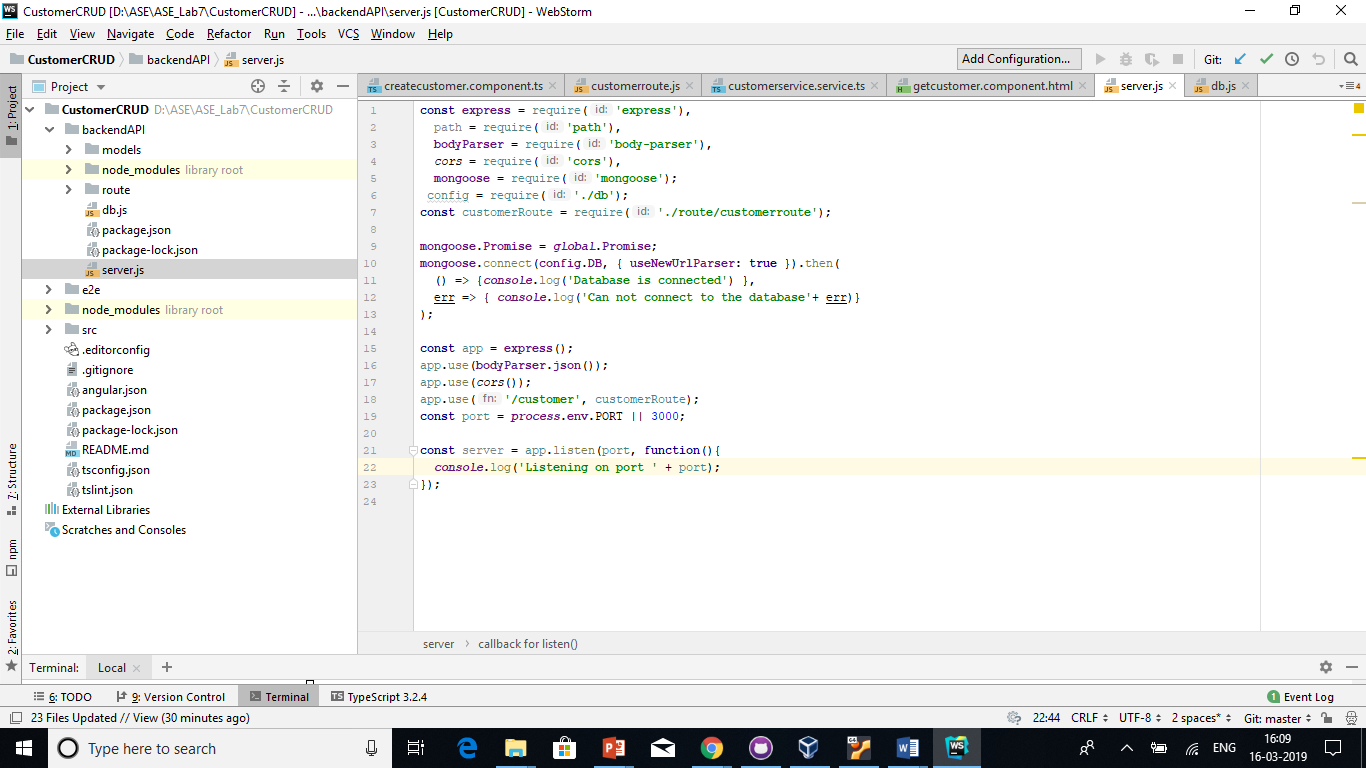
**Step-10:** Now, inside the **api**folder, create one file called the **server.js**file where in it contains the code to connect with port number 3000



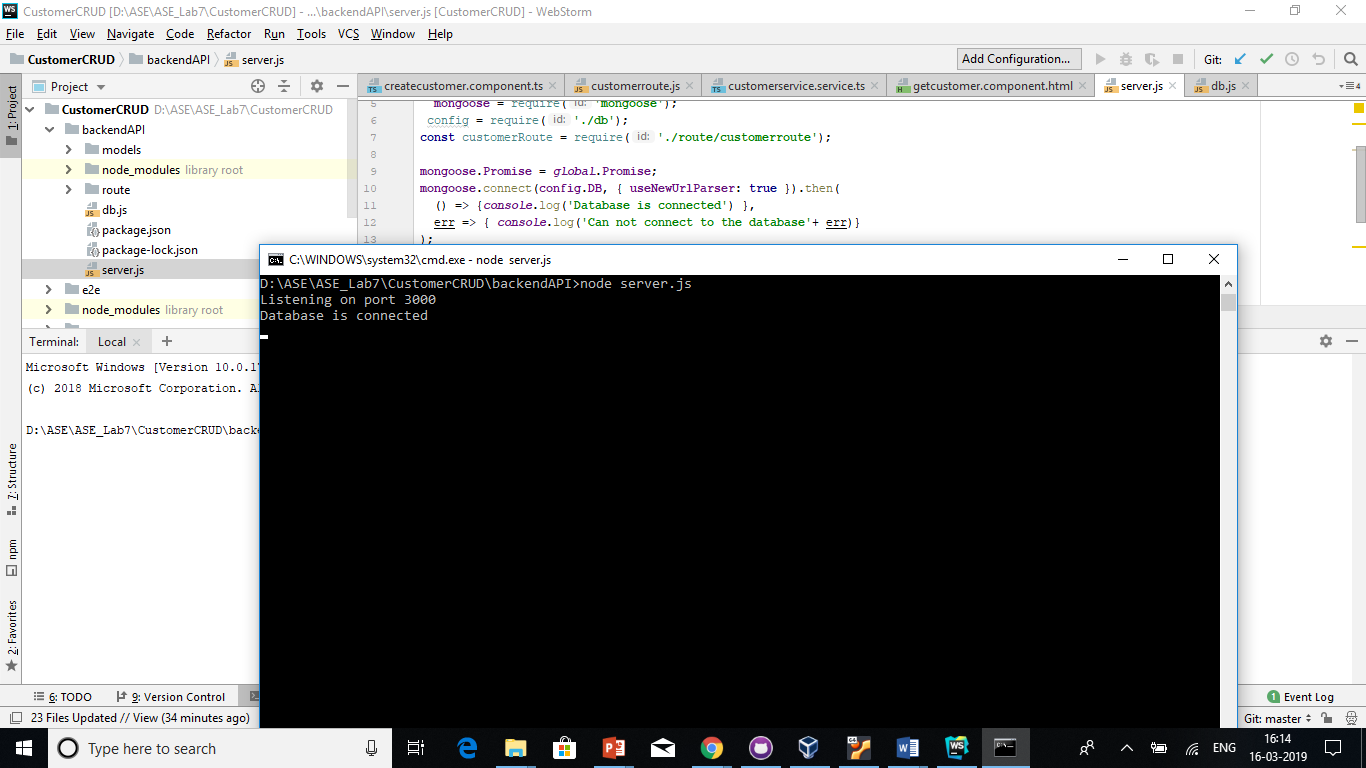
**Step-11:** Next thing is to connect MongoDB database with our node.js application.To achieve that we have created one file called **db.js** inside **backendAPI**root project folder. Written the following code inside the **db.js**file.



**Step-12:** Written the following code inside the**server.js**file to connect our **MongoDB**application to the **Node.js**server.



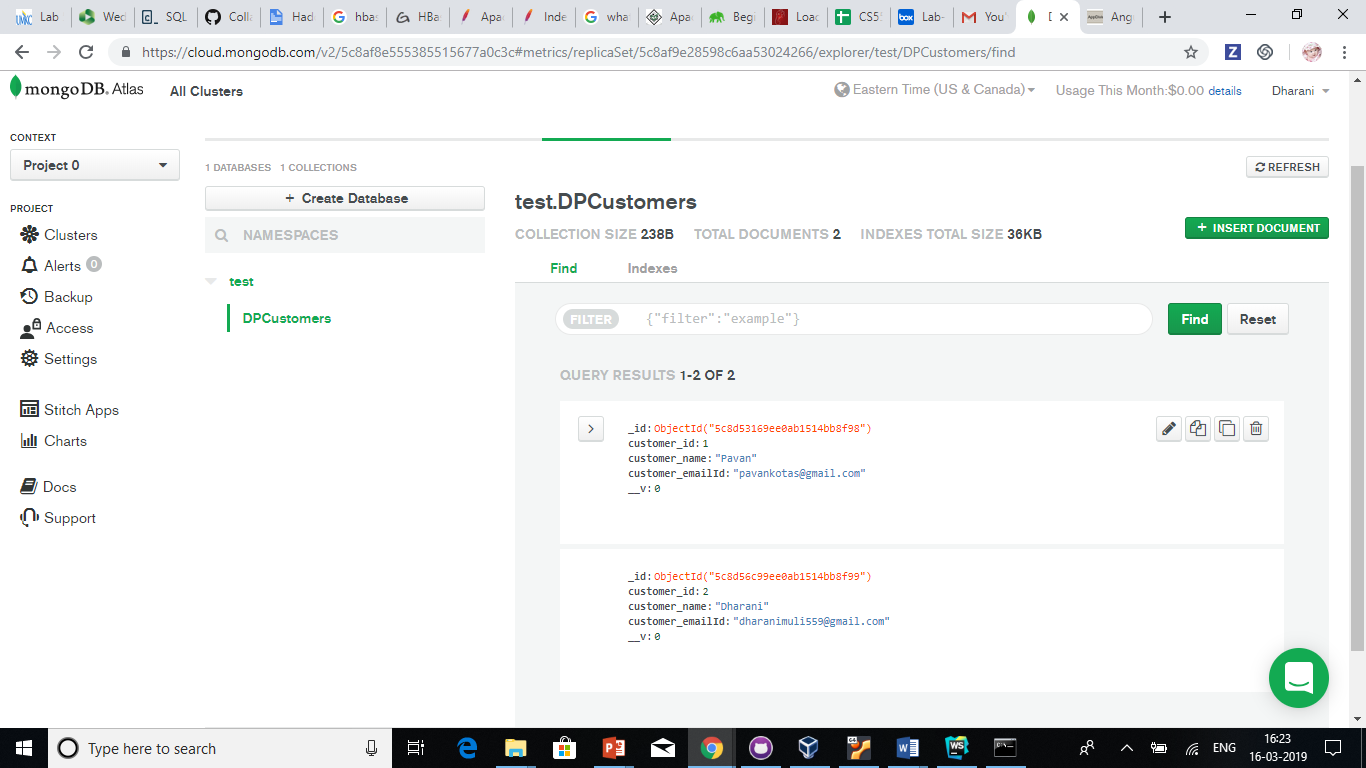
**Step-13:** Save our files and go to the terminal and start the node server to check weather we are able to connect to node and database or not



**Step-14:** Now created 2 folders inside the backendAPI root folder called route and models. In the models we have created custoemrmodels.js file and in the route customerroute.js file and we have defined our schema for the customer collection. We have three fields called **customer\_ID, customer\_Name and customer\_EmailID.**

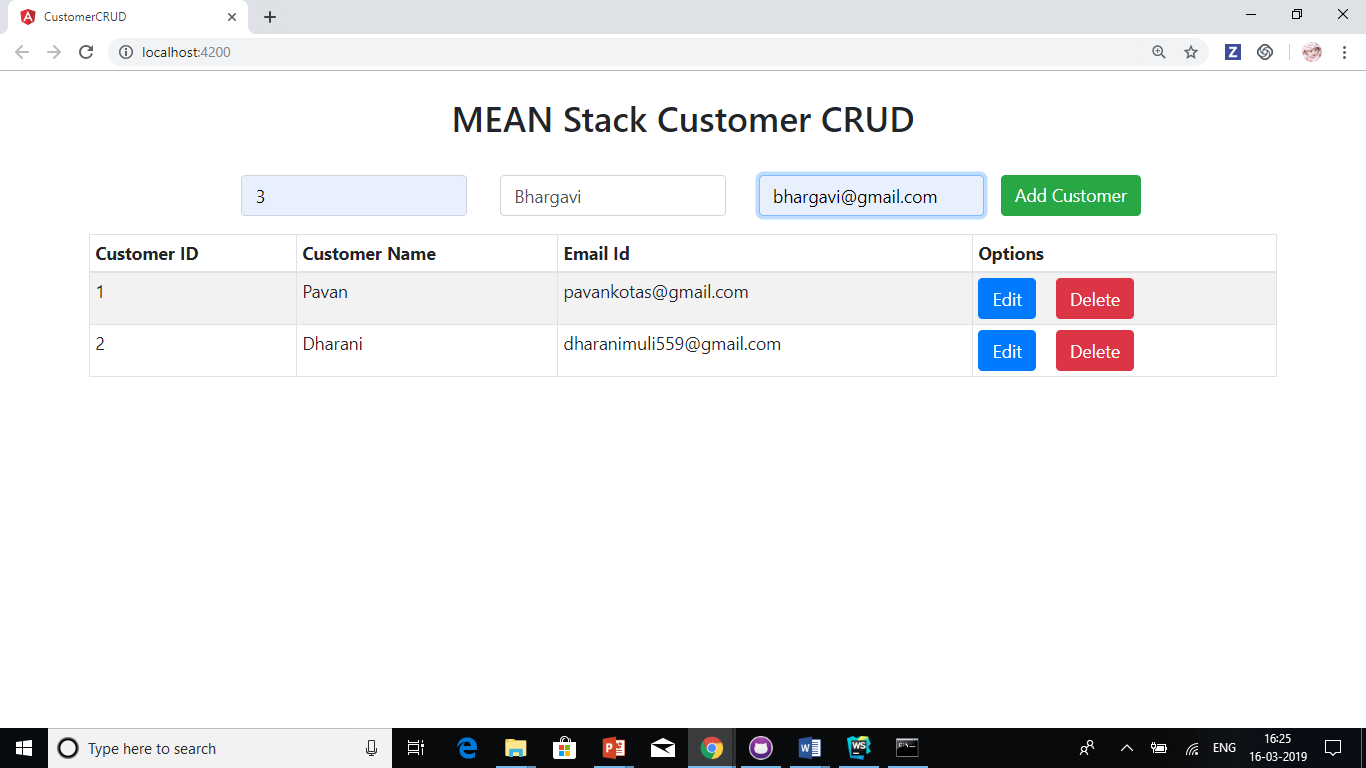
And created the CRUD code inside the **customer.route.js**file.

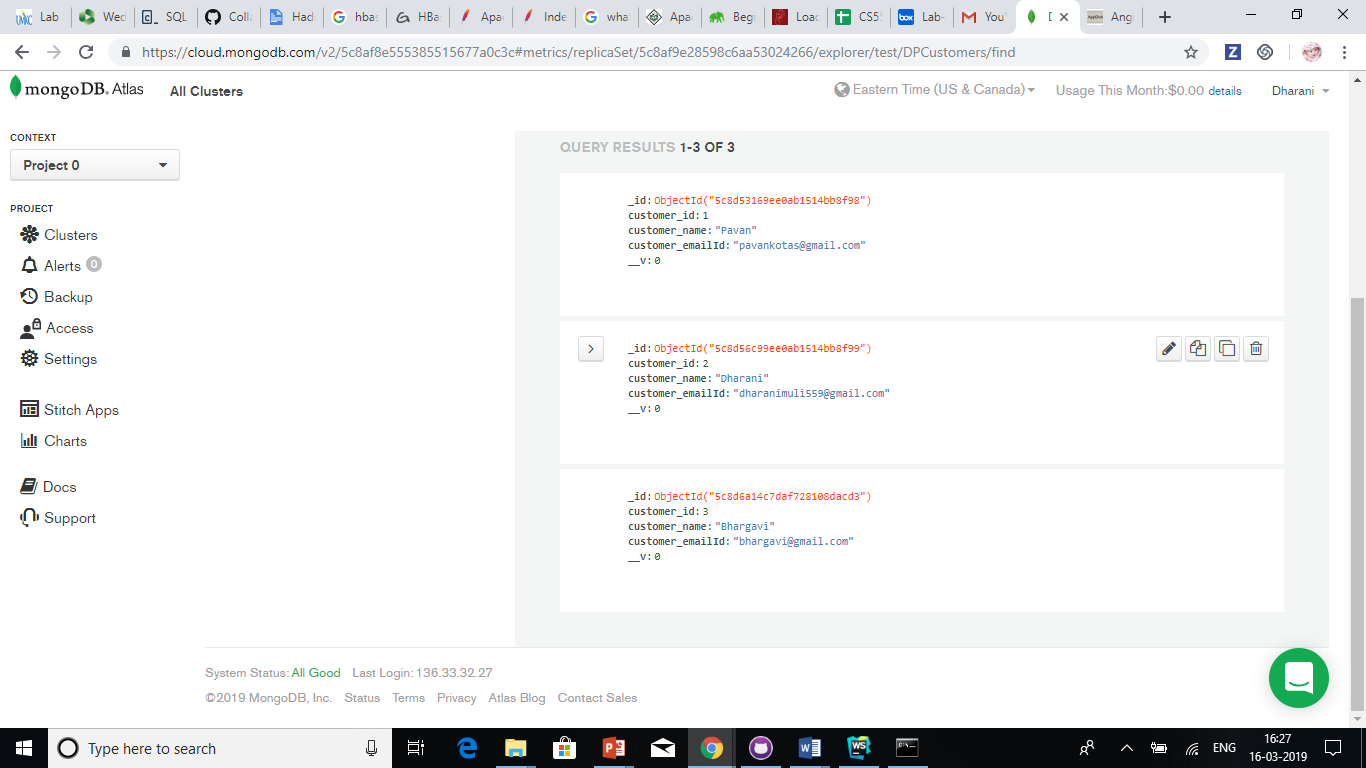
**Step-15:** We have created code under components.html files to display the data on the frontend and after running our code using command “ng serve -o” we can view our UI page where in when user add data in the input fields user can view data is getting into table, that says it is also getting added in MongoDB successfully.



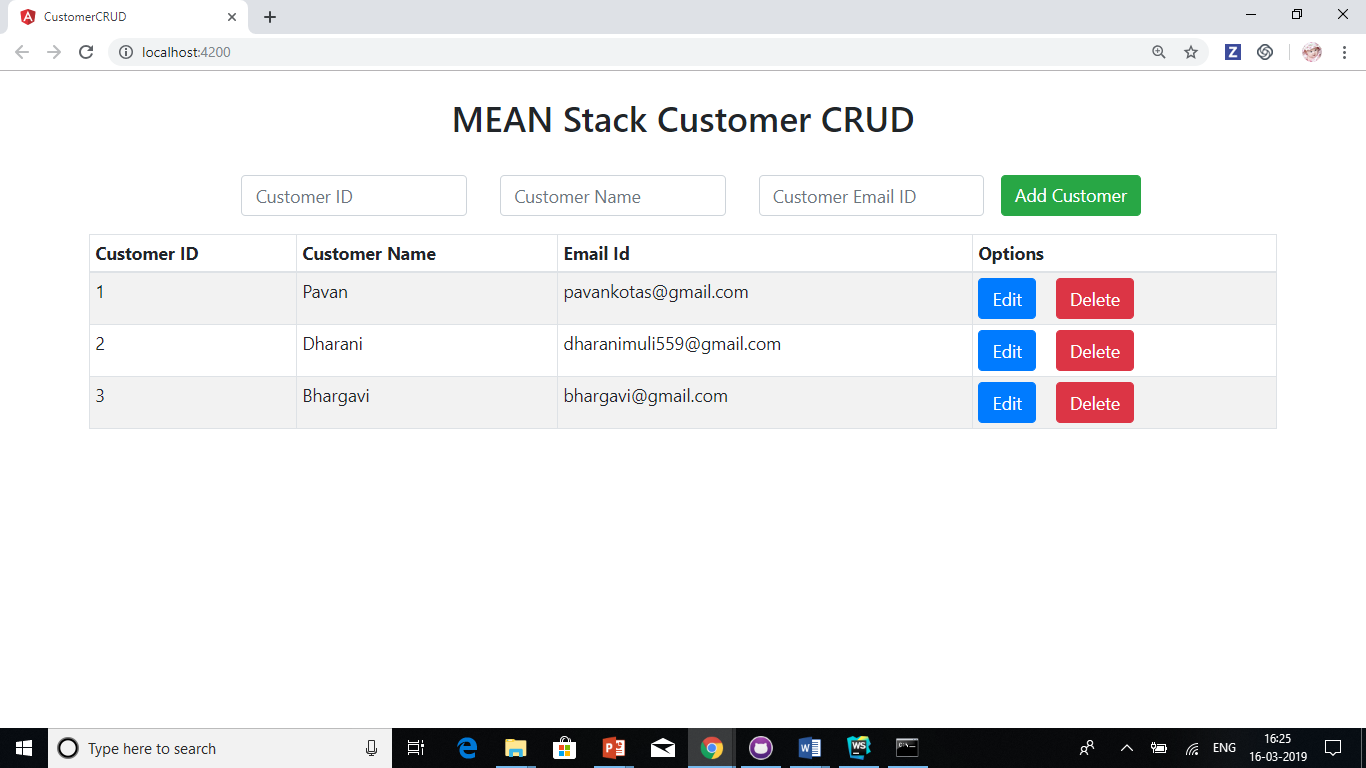
**Sample output**

1. Adding Customer

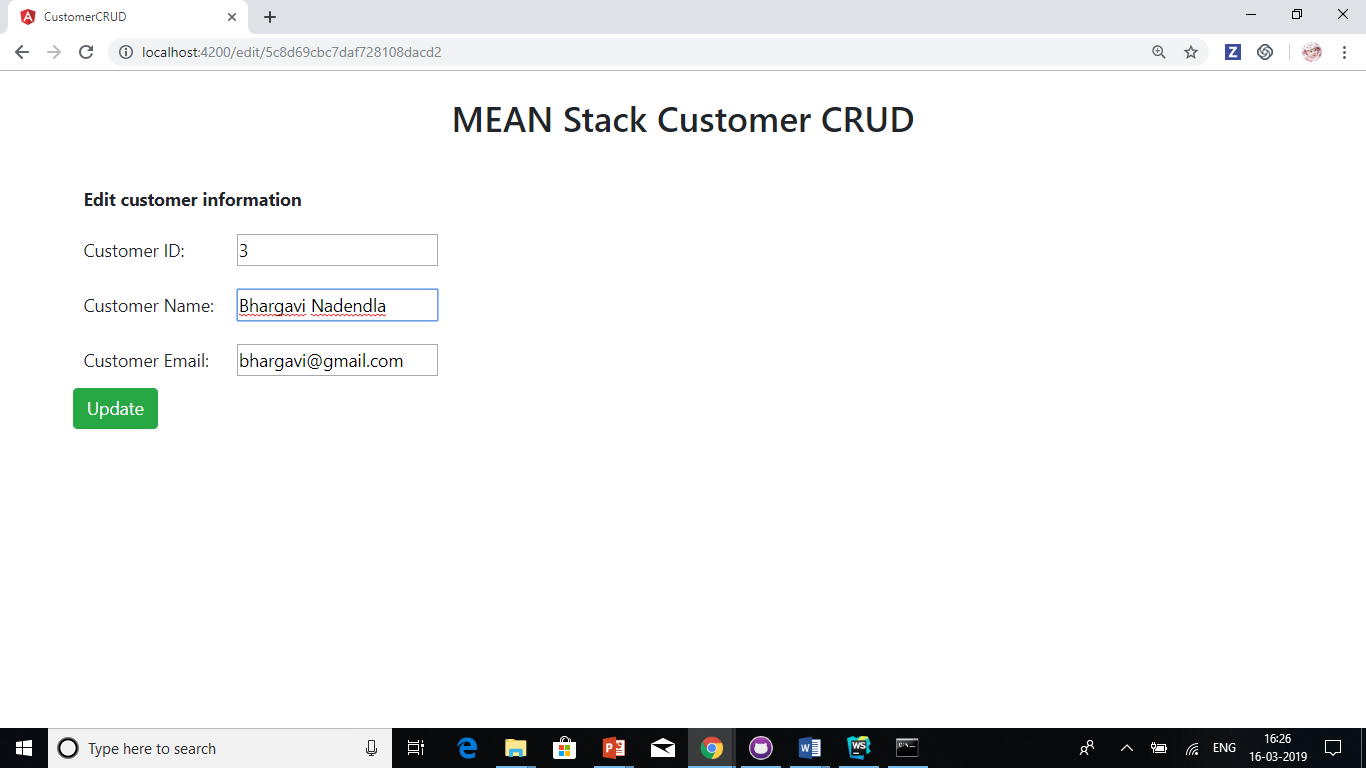


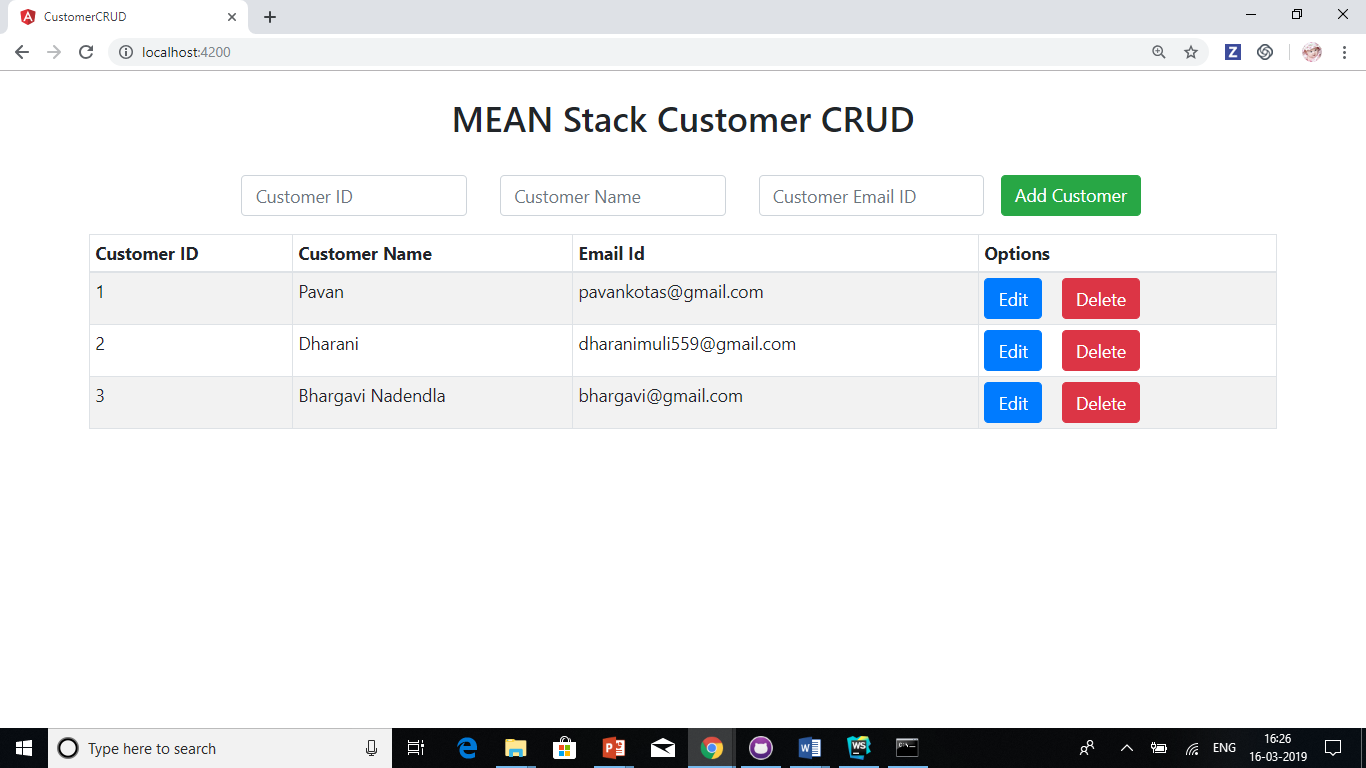


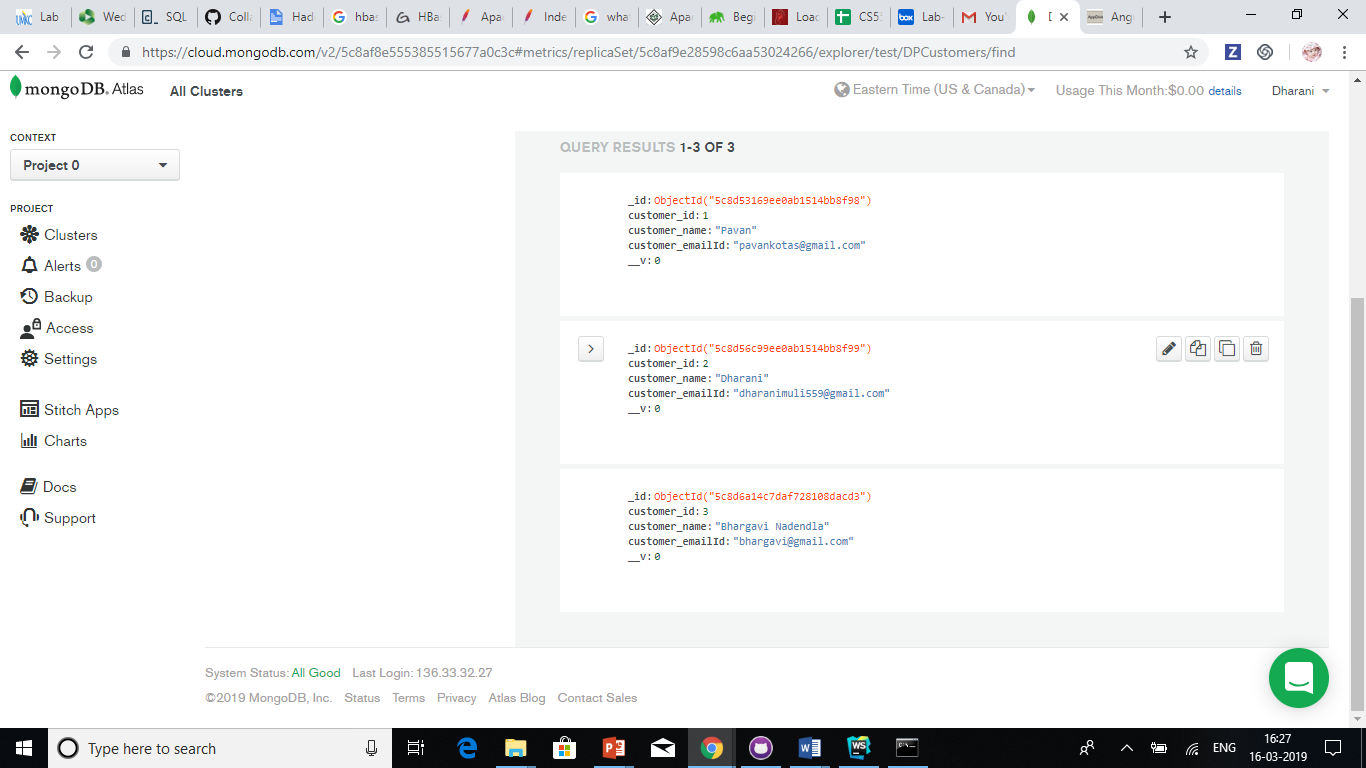
1. View the Customer in the table



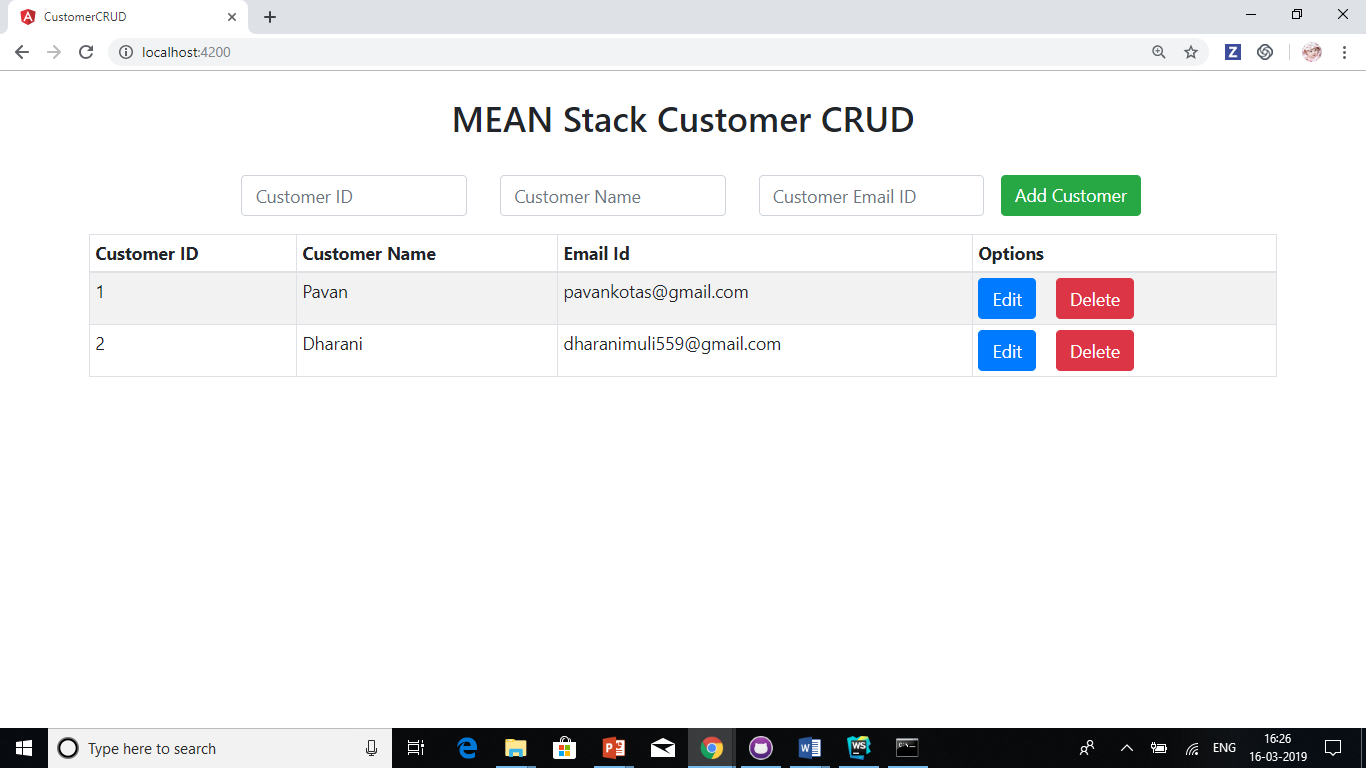
1. Editing the customer

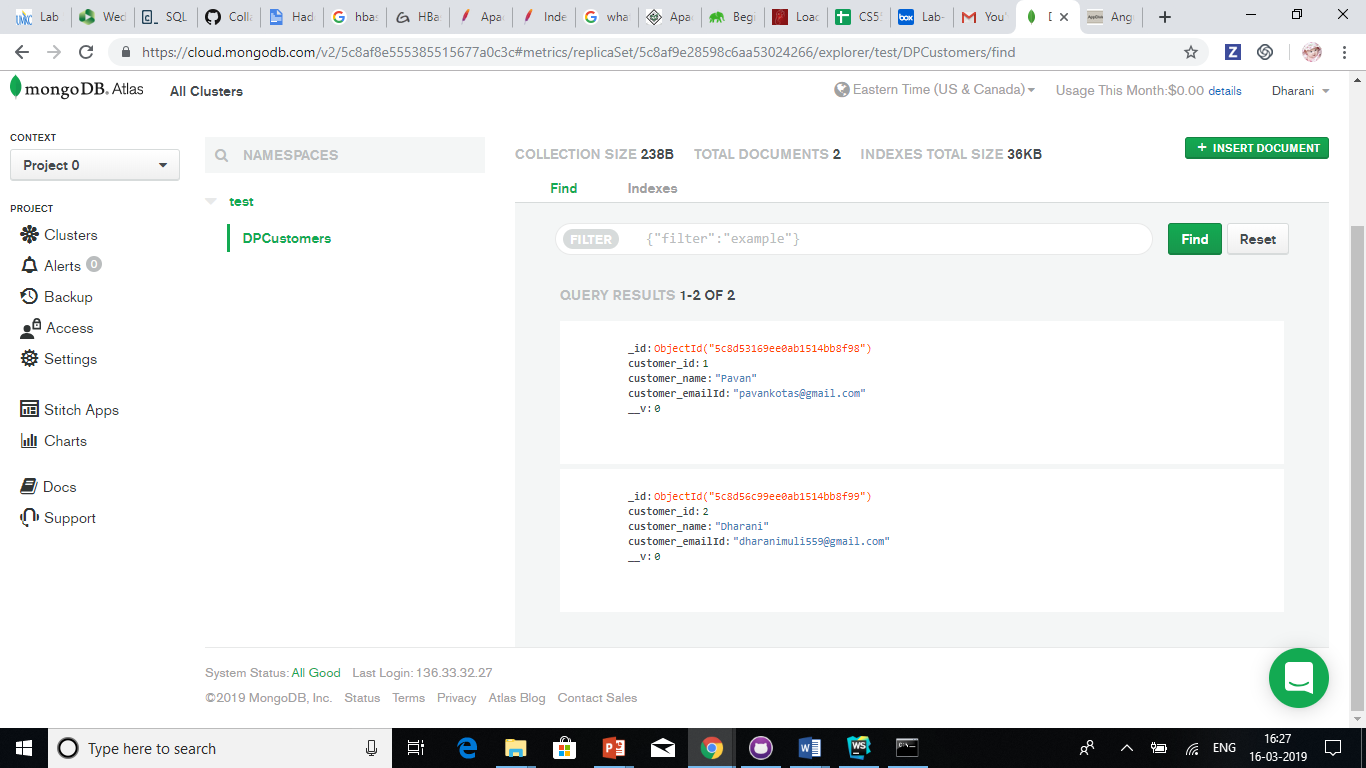






1. Deletion of customer





Issues/Limitations:

1. Faced few problem while connecting and doing CRUD operations in Mondo DB but later we successfully resolved all the problems and able to complete the assignment.
2. We have not much focused on the designing UI because main aim of this project is to build CRUD operations using MEAN stack with the basic frontend. But we have done basic good looking UI not the best looking.

Team Contribution

Dharani:

1. Created MongoDB account and established its connection
2. Implemented “Add” and “Listing” functionalities and also created interactive commands for the same.
3. Contributed in the creation of Wiki page, report and lab submission.

Chakra Pavan Kumar Kota:

1. Helped in resolving issues while connecting to MongoDB and saving data.
2. Implemented “Update” and “Remove” functionalities.
3. Contributed creation of Wiki page, report and lab submission.

Conclusion:

With this exercise we are able to understand the way CRUD operations can be performed in Node.js. We are able to analyse the previous source code provided and reapply the skills while implementing the CRUD repository for Customers. This javascript based node.js approach to access files, update them seemed to be very exciting which is otherwise impossible in plain JavaScript world. This also helps us in incorporating the skills in the final project whose backend is Node.js.