

Online Host, Visualization, and Deployment

**Submitted by**

**Vamsi Draksharam: Class id -9**

**Goutham Gandreddi: Class Id -11**

**Premchand Lingamgunta: Class id - 15**

**Project Goal and objectives:**

**Motivation:**

Abundant models and iterative tests on the models, raised the need for a system to capture and host a model, visualize the results and sharing the trained models online becomes a crucial step of the overall deep learning lifecycle.

**Objective:**

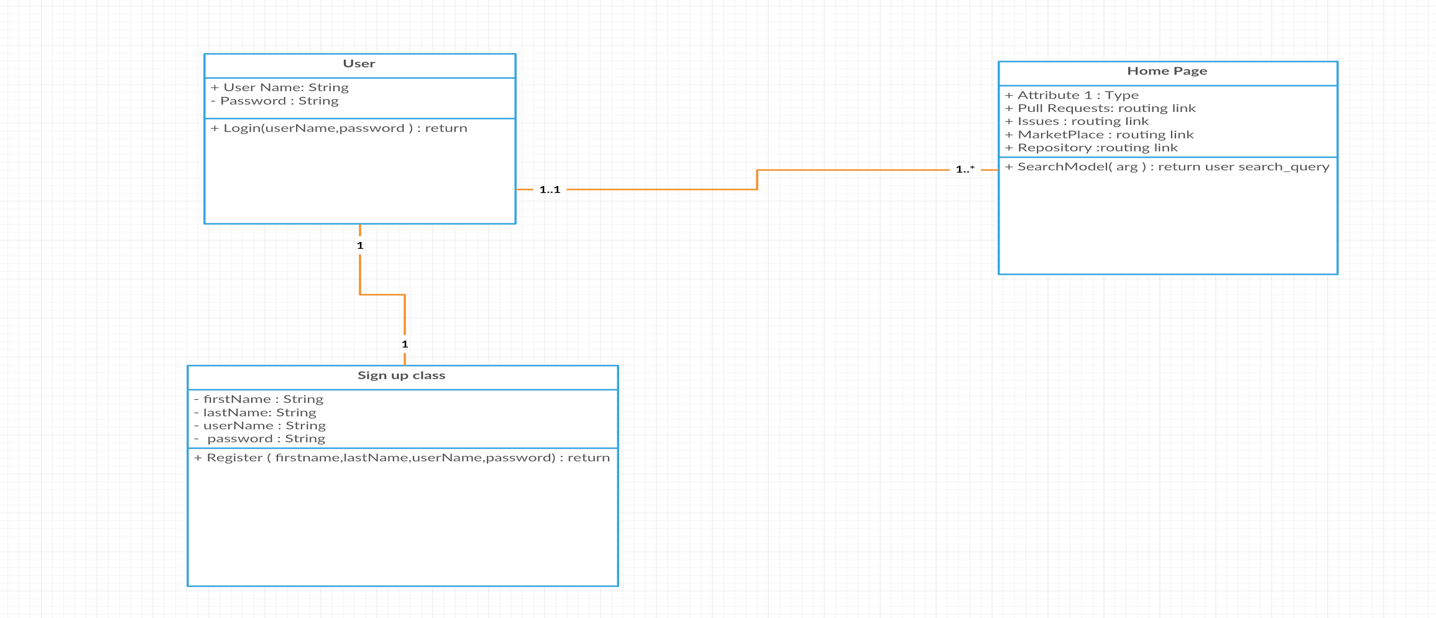
As mentioned above, ‘online host, visualization and deployment’ gives the user a platform where user can upload or download the models or pretty much anything. User can view any model as a separate module and can comment and rate the model.

We are following the MEAN stack approach using Angular as front-end application framework, to create the views for user interaction, Node js and Express to handle the business logic and establish the connectivity to database, for user validations and retrieving the details regarding the models. The Discussion form for each issue is going to have a thread of comments and other features which are planning to be implemented.

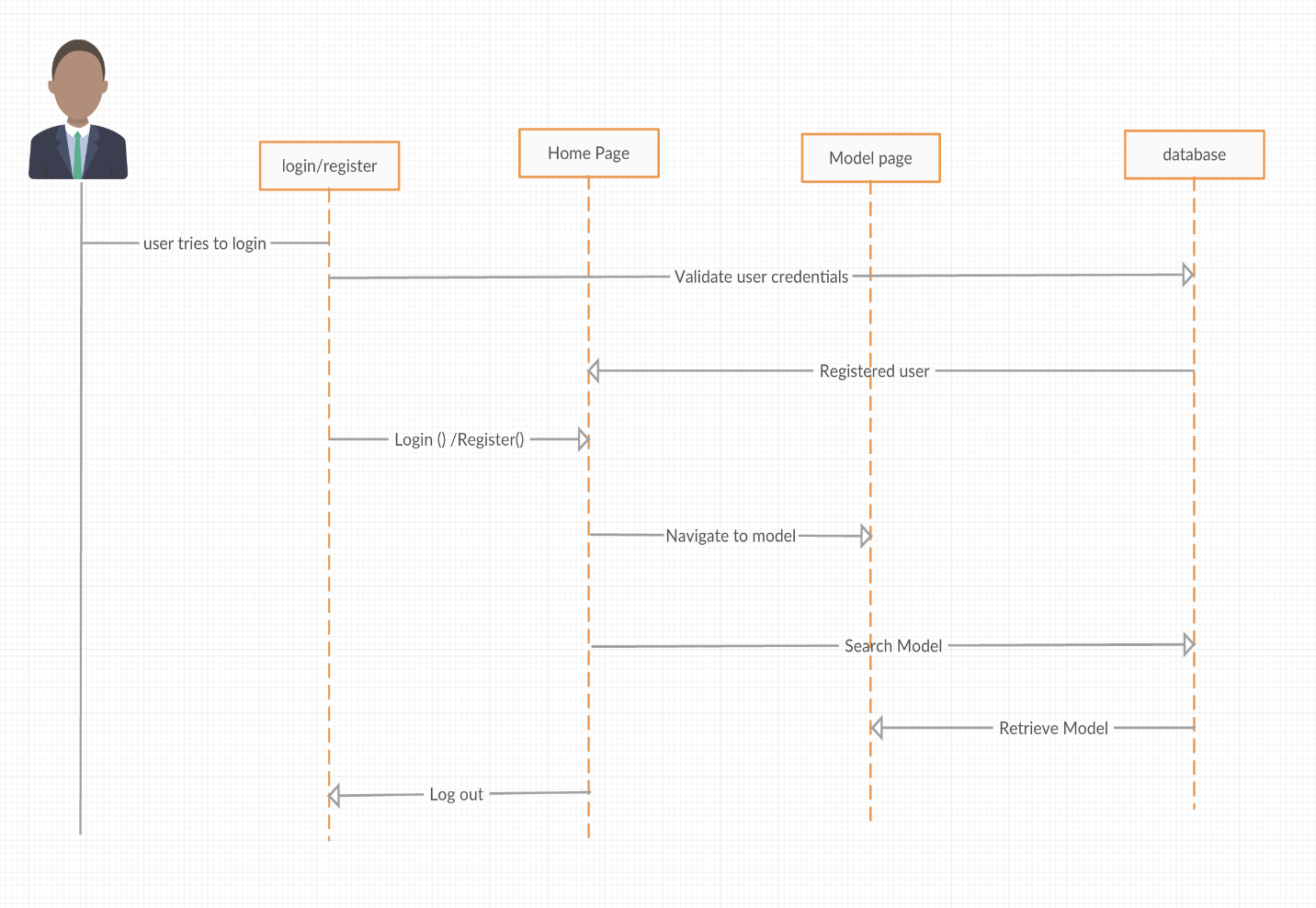
**Second increment Report:**

**Architecture Diagrams :**

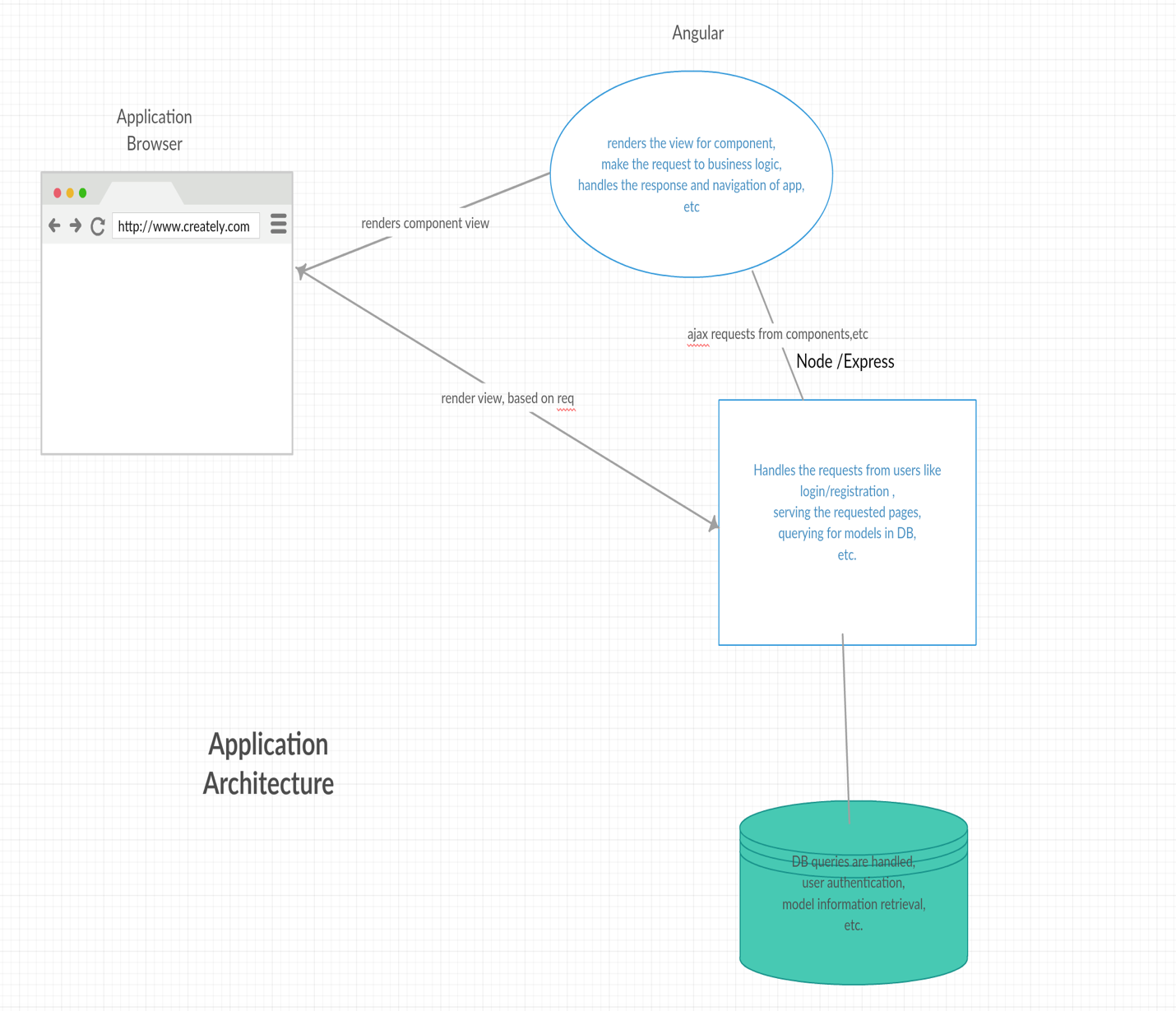
1. **Class Diagram:**



1. **Sequence Diagram:**



3.Architecture Diagram:



**First Increment**

**Singup & Login Page**

Premchand Lingamgunta – Design and Implementation of the Home page, registration page.

Design and created the backend storage of the user details after login.

Goutham Gandreddi- Home page layout and routing between log-in,registration pages to homepage and showing user information.

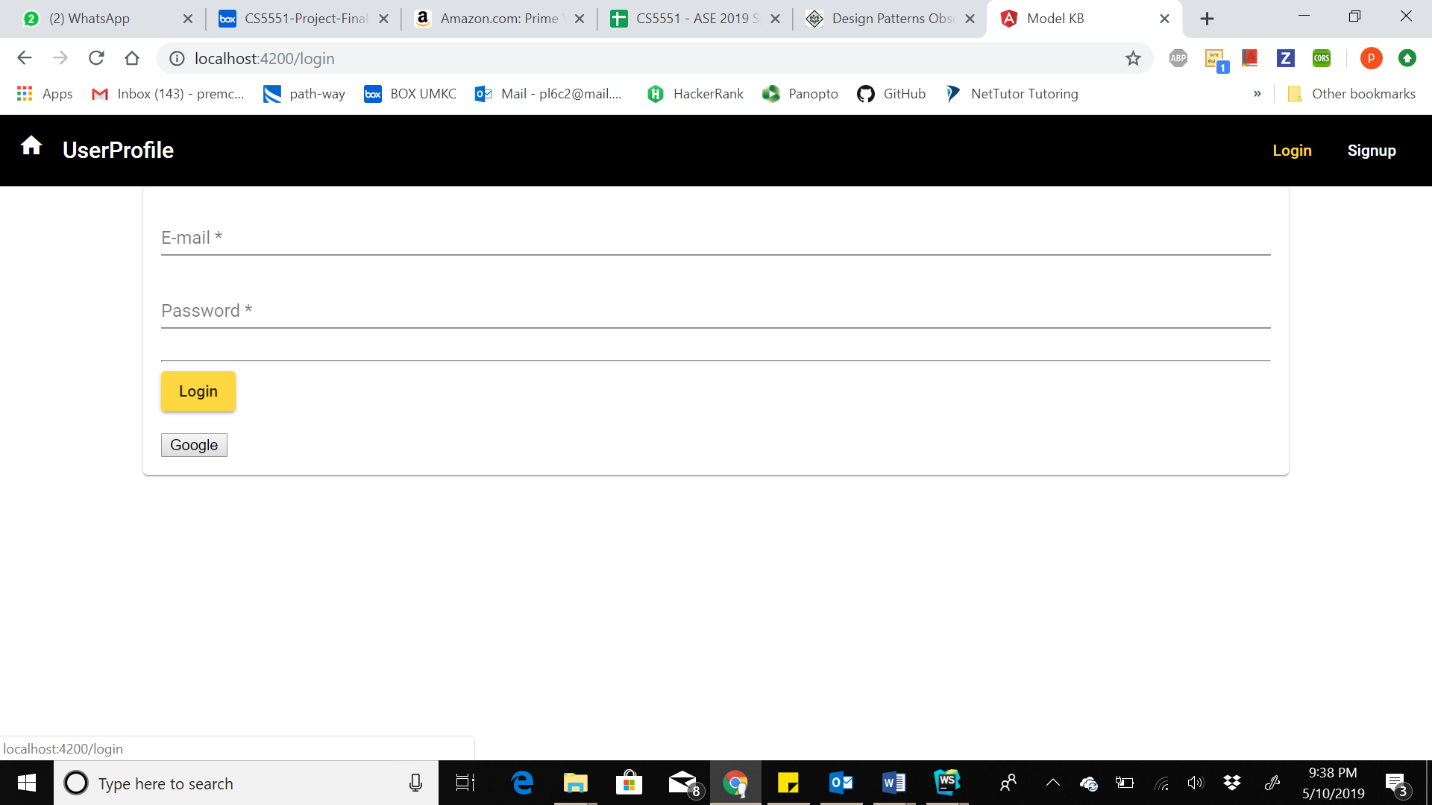
Vamsi Draksharam - Designed the html and layout for the Login and registration page

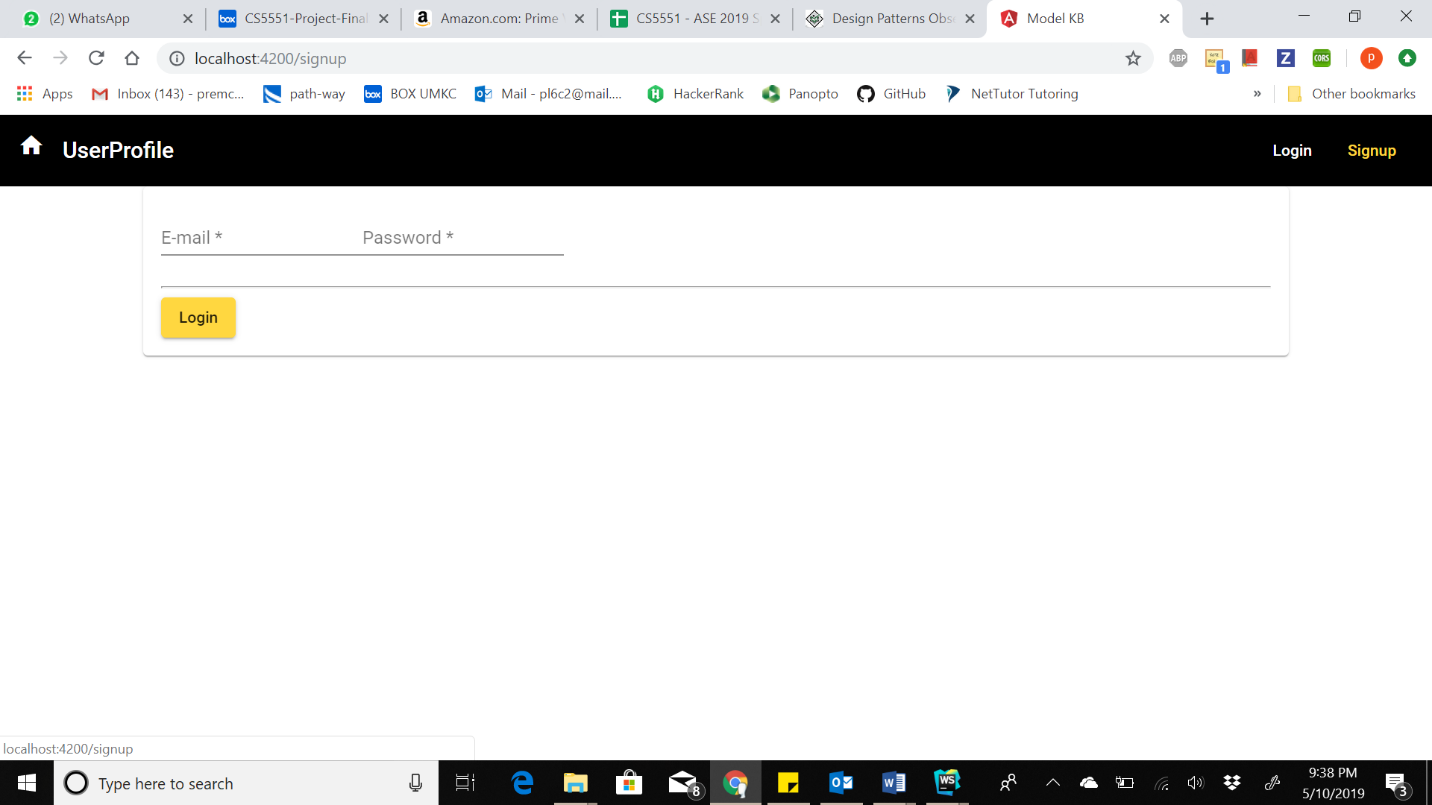
* User need to singup first to utilize the features of our application
* When user signup the details will be stored at backend. Only unique users are allowed to use our application.
* Now user can login into the application with the correct credentials other wise user will get a pop-up saying “enter valid user credentials”.
* When we click on login button we will navigate to home page.

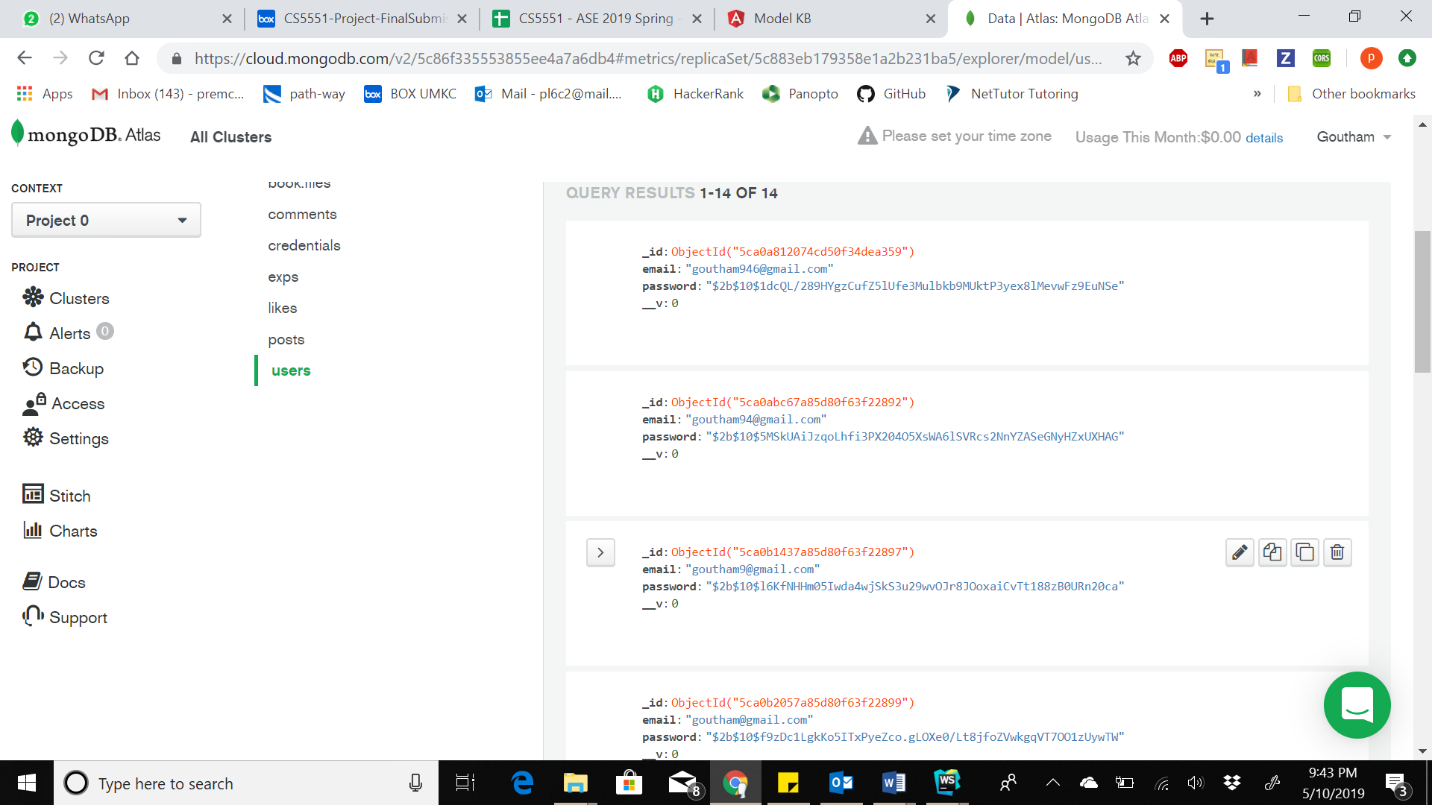
Signup.component.html -> Signup.component.ts -> AuthService -> backend

Login.component.html -> LoginComponent -> AuthService -> backend

Homepage.component.html -> HomePageComponent.ts







**Second Increment Implementation:**

Vamsi Draksharam - Login page solved issues: included proper validations, like if the user is not entering correct details and there is a mismatch in the user name and password, the users gets an alert to re-enter correct details.

Premchand Lingamgunta  & Vamsi Draksharam – Design and Implementation of the registration page.

Premchand Lingamgunta – Design and Implementation of the User Profile.

Goutham Gandreddi- Implemented the login page with encryption and high security, used token-based system for data communication between client and server.

Goutham Gandreddi - Designed the Repository page layout and view in front end, creating the model page and upload a image to repository.

**Backend Data Creation and manipulation:**

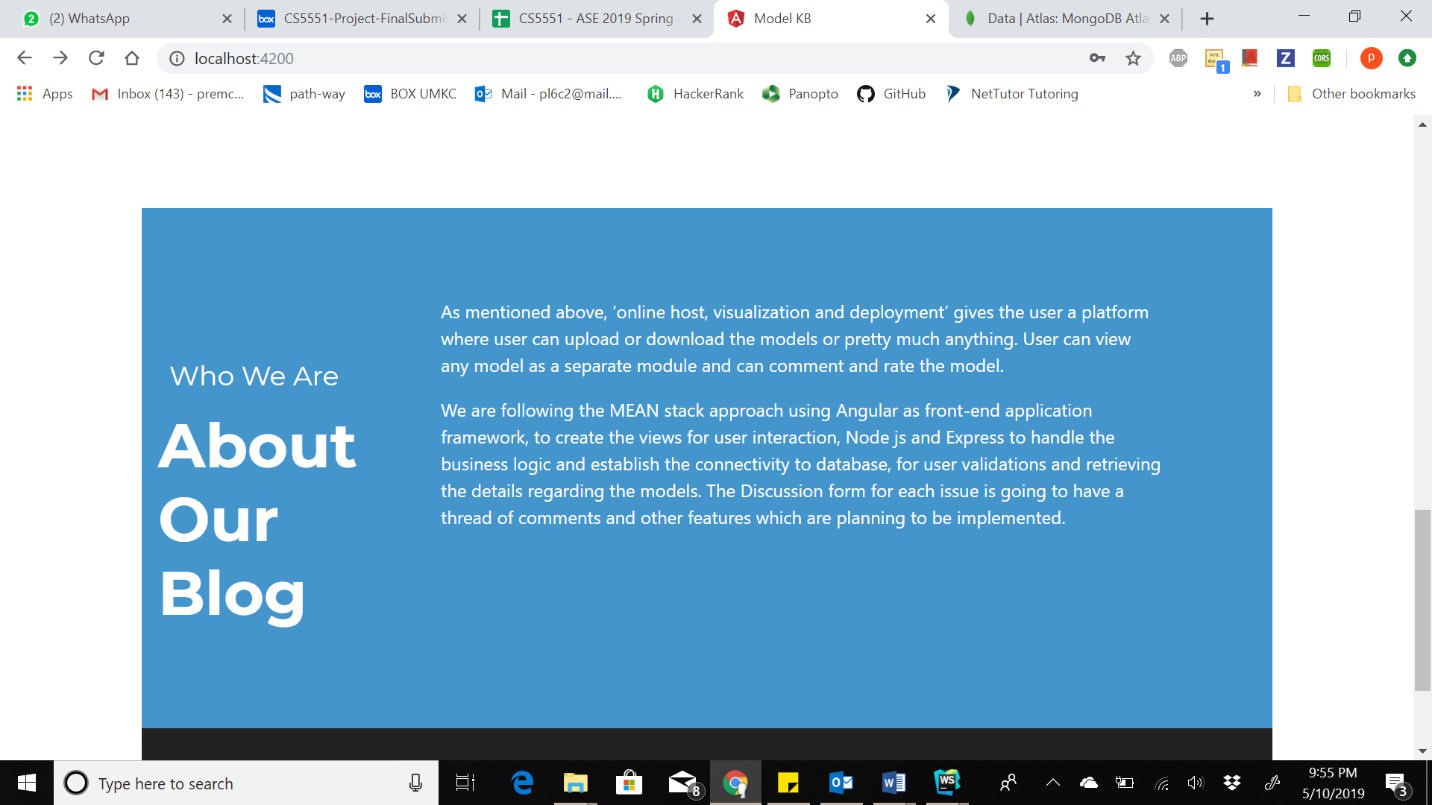
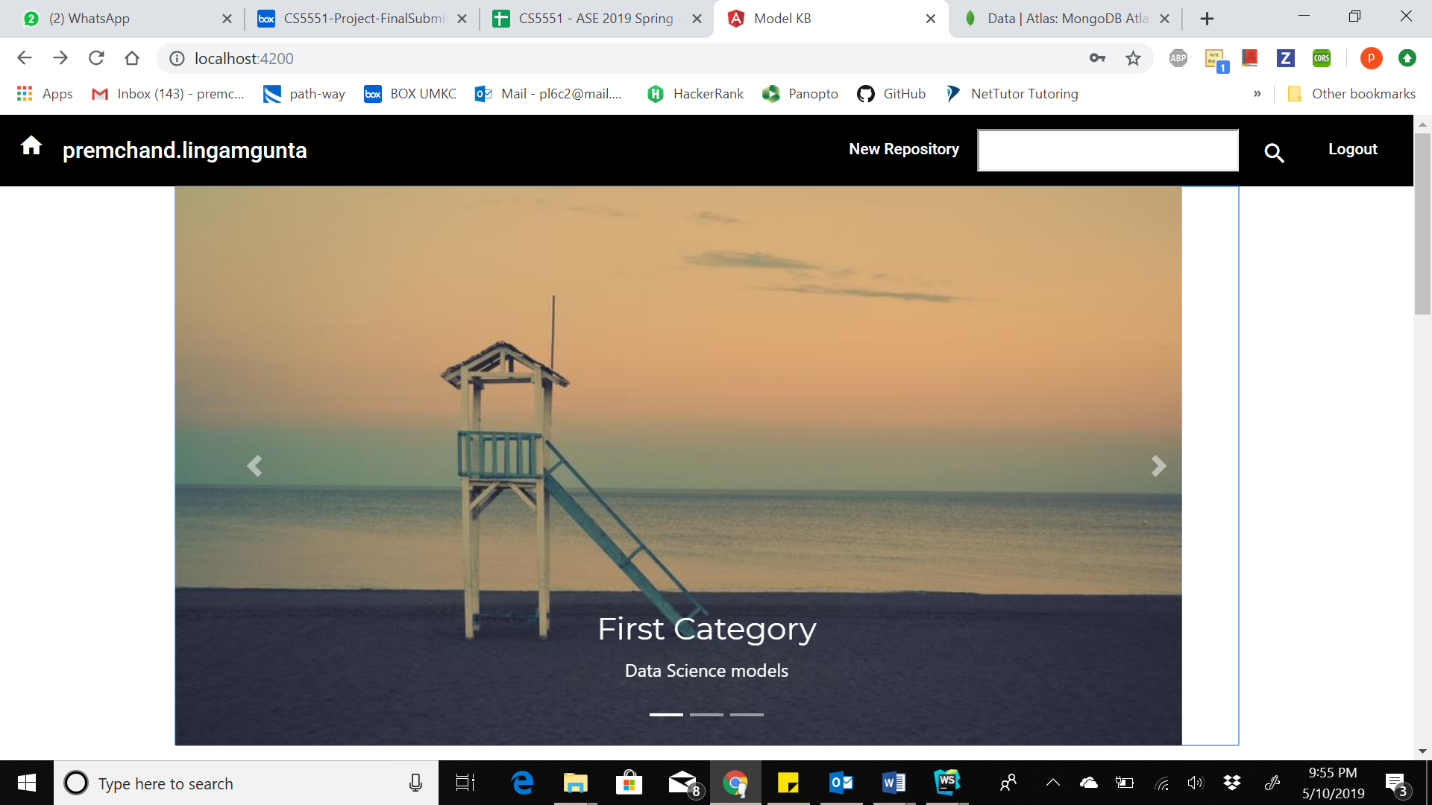
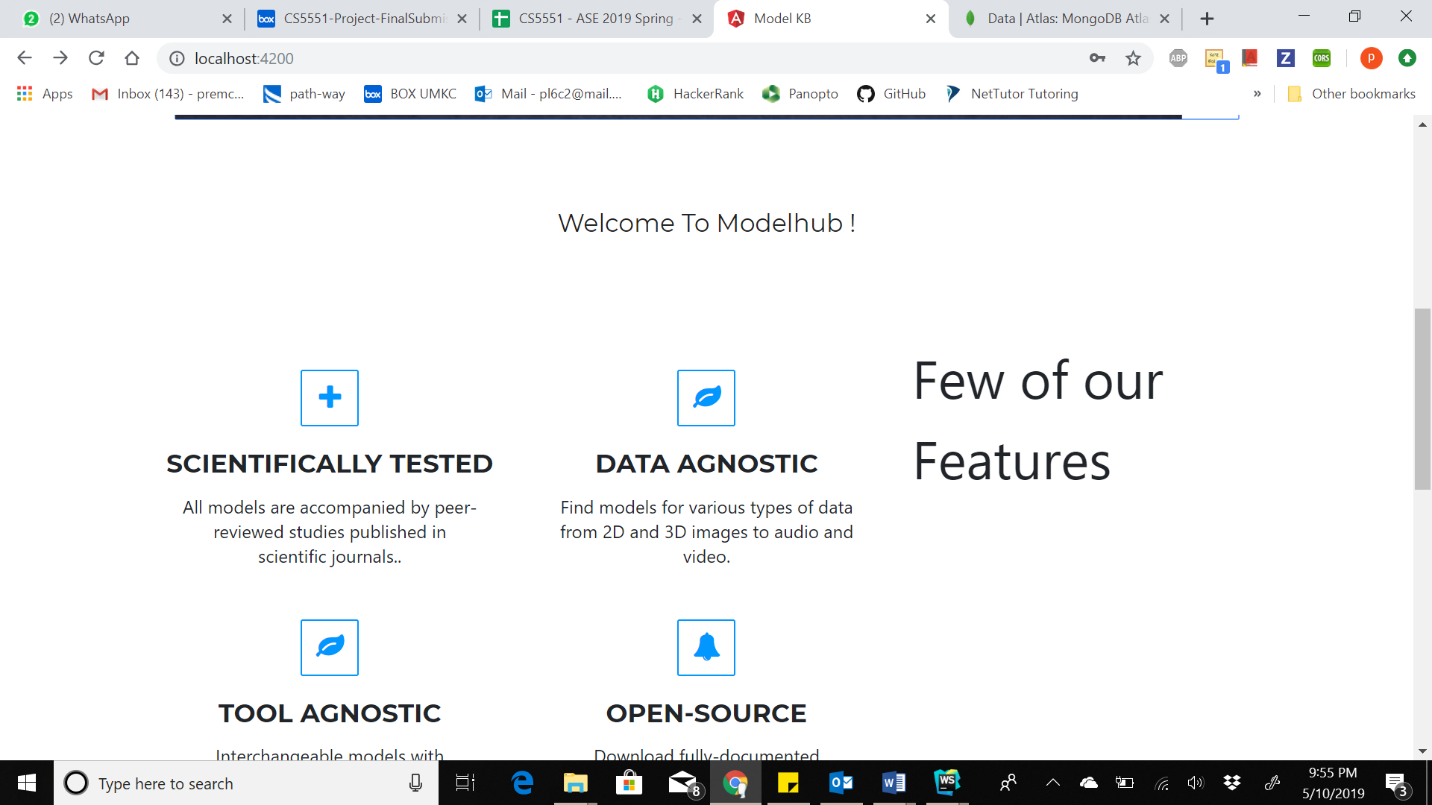
Premchand Lingamgunta – Design and Implementation of the registration page.

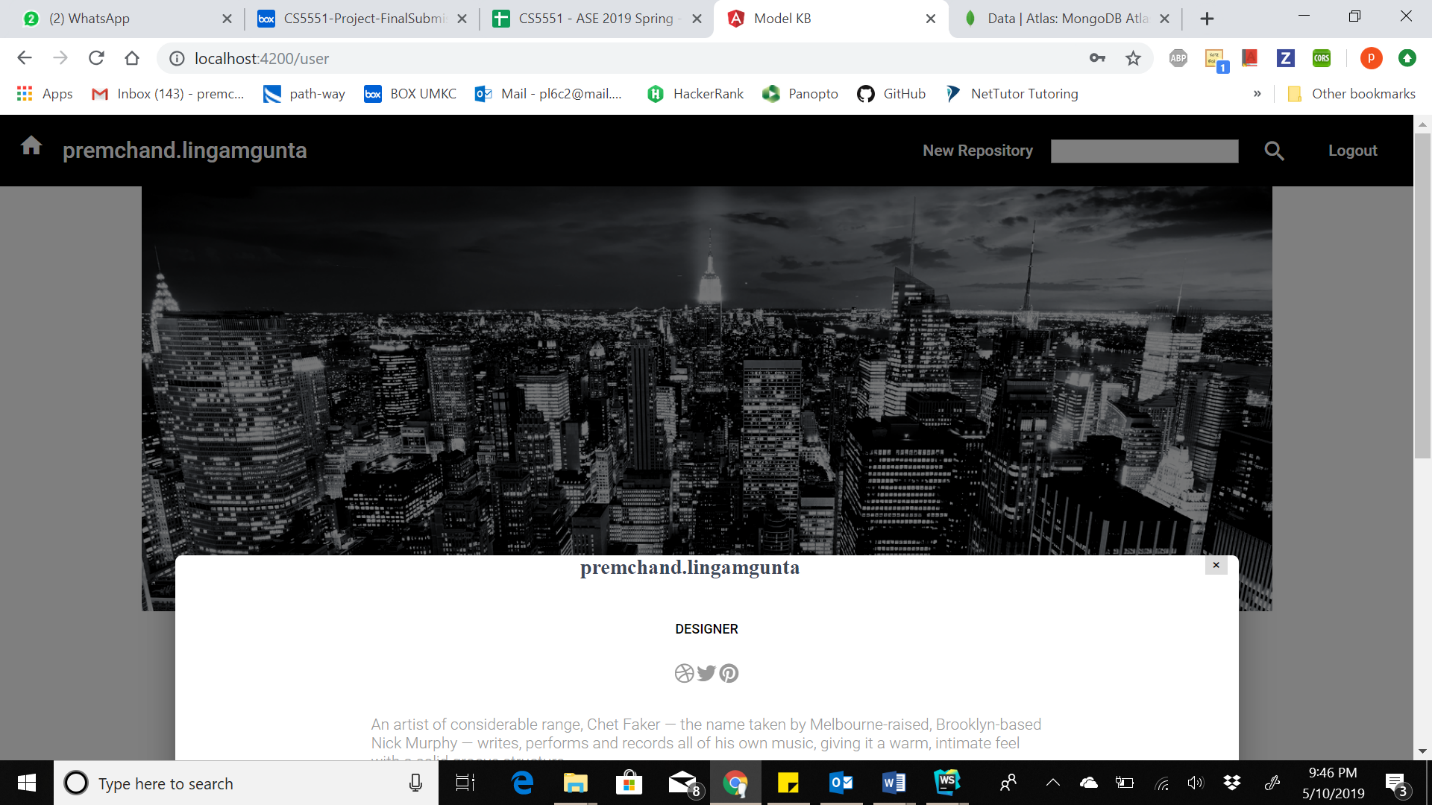
Vamsi Draksharam – Home page design and Implementation for backend.

Goutham Gandreddi - Designed the Repository page layout and implemented the storage of repository details in mongo dB.

UserProfile**.**component.html -> userprofile.component.ts

* In this increment we have created user profile, menu bar and Home page.
* In user profile you can find details like models he has uploaded. Menu bar consists of several links username and New Repository and search bar and logout.
* When you login the user name will be appear on left side of the menu bar
* Before login we see only login and singup links when user logged we will have those links on menu bar.





**Third Increment Implementation:**

**Upload files for a repository:**

Premchand Lingamgunta – Design and Implementation of the User profile page.

Premchand Lingamgunta – Backend Data Creation and manipulation for User profile.

Goutham Gandreddi – developed features like “User details and editing features” are only enabled when login is success.

Goutham Gandreddi – listing the repositories or models in the user profile page

Goutham Gandreddi – Backend Data Creation of tables and manipulation for repositories or models data.

Goutham Gandreddi – Implemented the filesystem-based storage for the model files after upload.

Vamsi Draksharam – improved the login page and registration page layout.

Userprofile.component.html -> userprofile.component.ts -> AuthService -> backend

Post-create.component.html -> post-create.component.ts -> postService

Post-edit.component.html -> post-edit.component.ts -> postService

Post-delete.component.html -> post-delete.component.ts -> postService

* Included uploaded feature to add files for the repository and they can be viewed.
* Added like feature to the models.
* Implemented the filesystem-based storage for the model files after upload.





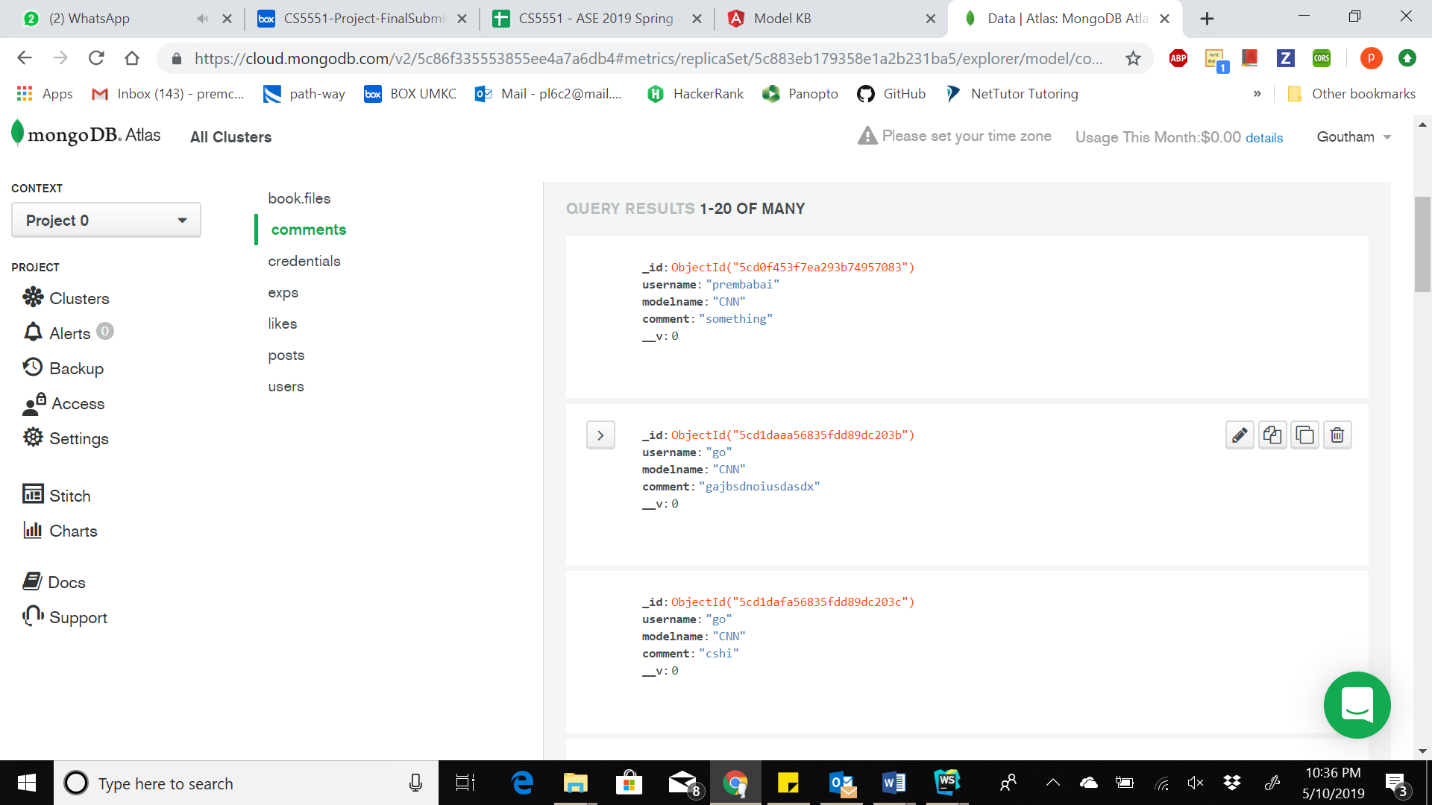
**Like Button for Models**

* This feature has the ability to like or dislike the particular model
* When user clicks on like button the details will be stored in backend
* And fetch response to front-end.

Like.compoent.html -> Like.component.ts -> LikeService

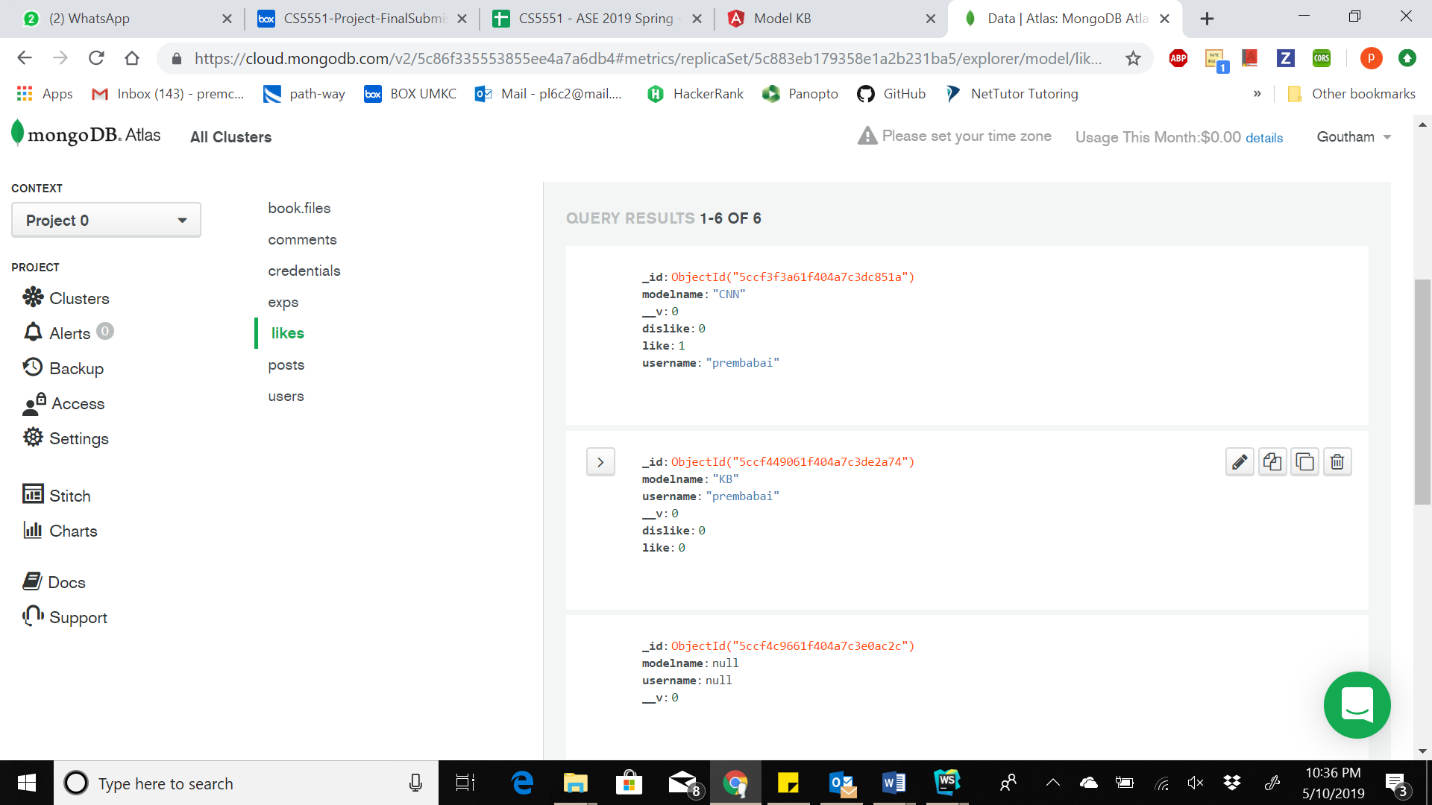
Component.html -> Component.ts -> CommentService

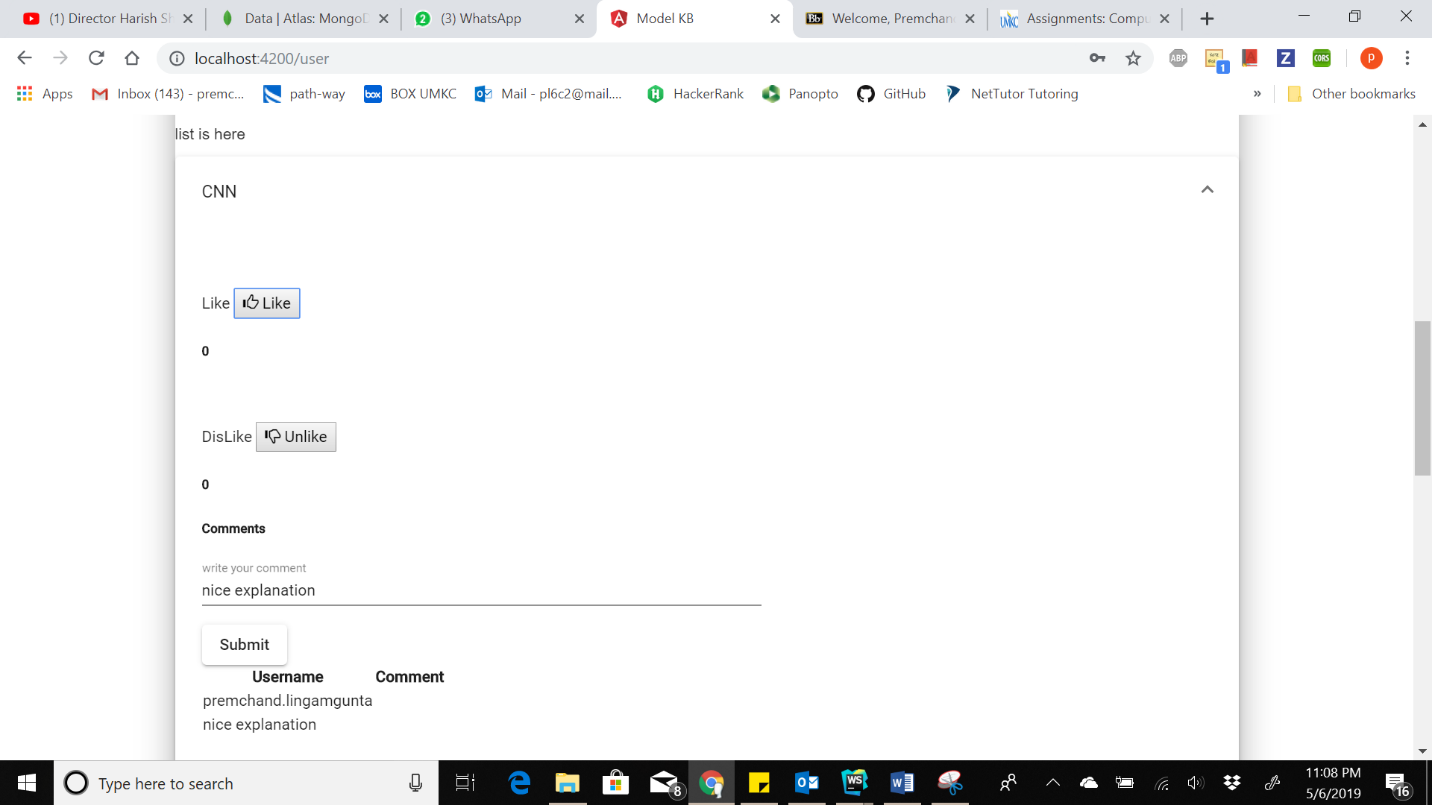




**Comments Section:**

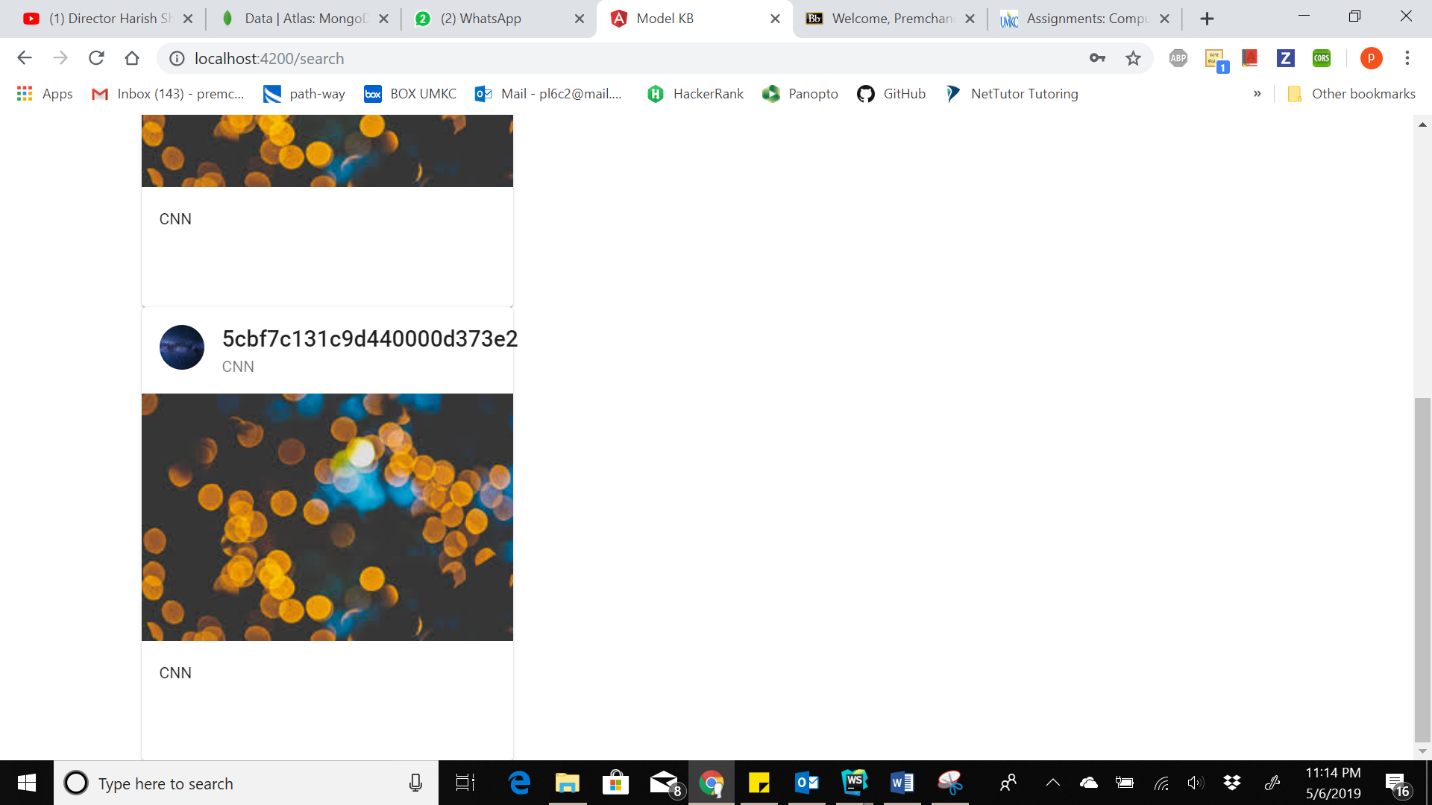
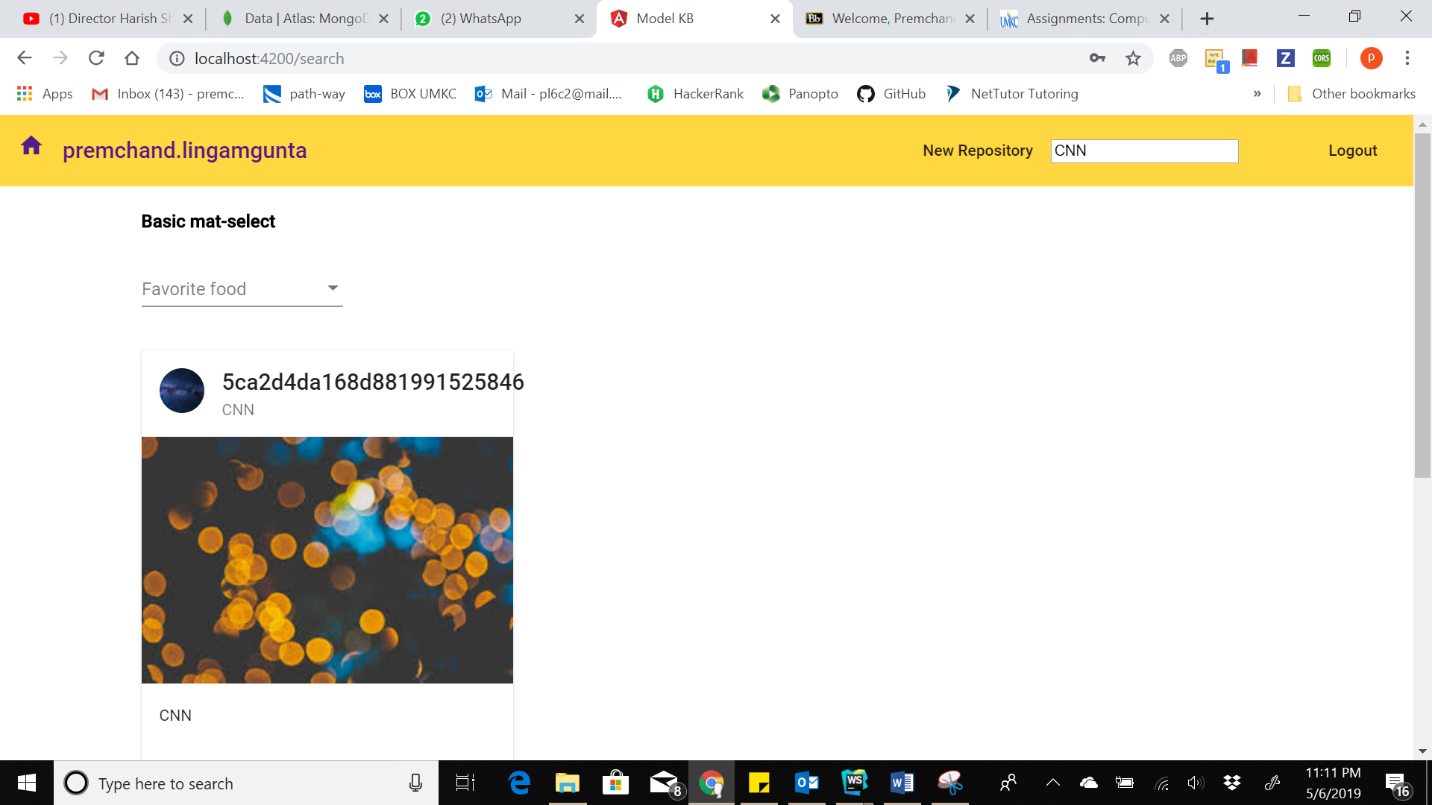
* This feature enables us to write comments for particular Model.
* When we write the comment and press submit the details will bestored at backend and get back with response**.**
* When user comment on the particular model. The comment will be added to that model and stored at backend.

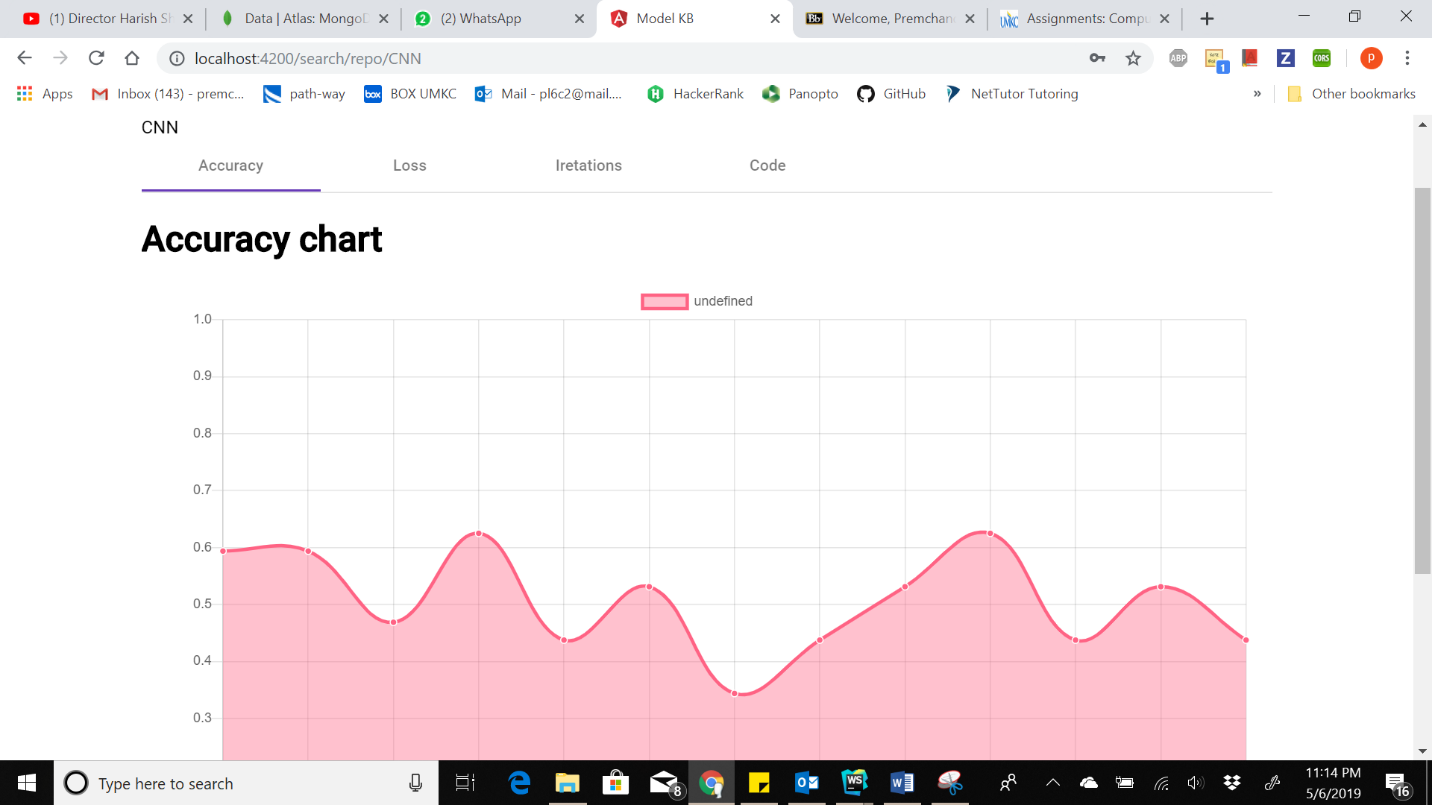




**Search for Model:**

* This feature enables us to search for a particular model which we have already. If we don’t have searched model it will display nothing.
* Here we have entered CNN model name and we got 2 models of CNN.
* If we click on mat-icon we redirected to visualization of that model**.**
* Search.component.html -> Search.component.ts -> AuthService

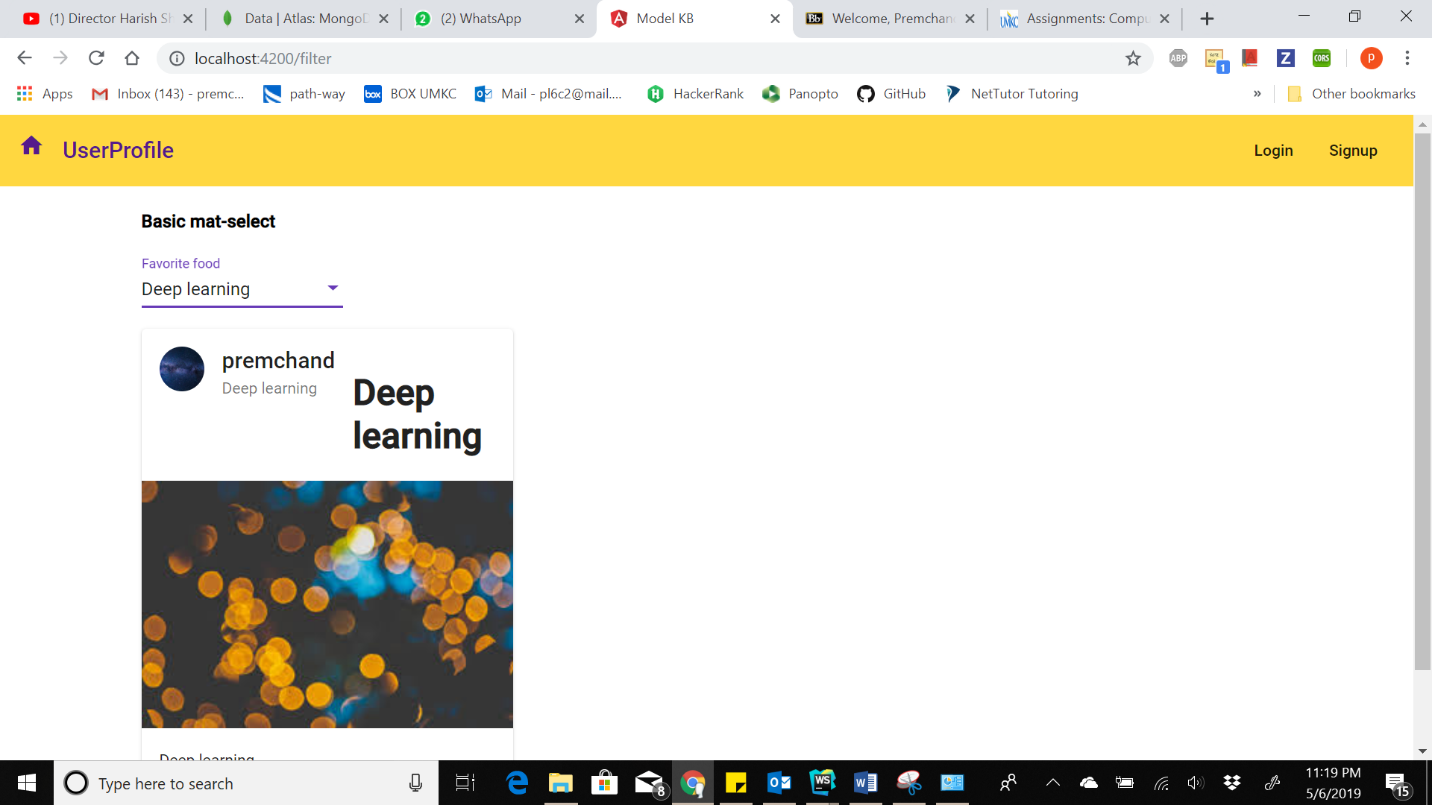




**Filter Feature:**

* This feature enables us to filter through the models which we have already in db.
* When we select our required model we will get back our results as shown
* Here we choose deep learning as our filter. So it will display models regarding our filter value.

Filter.component.html -> Filter.component.ts -> FilterService



**Project Management:**

**Implementation Status Report:**

**Work Completed:**

**Home Page implementation: –**

**Responsibility and Time Taken:**

* Premchand Lingamgunta – Design and Implementation of the registration page (5 days)
* Vamsi Draksharam – (5 days)
* Goutham Gandreddi – User details and editing features only when login is success (5 days)

**User Profile:**

**Responsibility and Time Taken:**

* Vamsi Draksharam – user info (3 days)
* Goutham Gandreddi – included repositories or models (5 days)

**Repositories:**

**Responsibility and Time Taken:**

* Goutham Gandreddi- Implemented the login page (4 days)
* Premchand Lingamgunta – implemented like feature (4 days)
* Premchand Lingamgunta - Searching files (4 days)

**Uploading Files:**

**Responsibility and Time Taken:**

* Goutham Gandreddi- Implemented uploading files(4 days)

**Searching and Filters:**

**Responsibility and Time Taken:**

* Goutham Gandreddi – Implementing the seach (2 days)
* Premchand Lingamgunta - Implementing the seach (2 days)
* Premchand Lingamgunta - Implementing the Filters (2 days)

**Backend Data Creation and manipulation:**

**Responsibility and Time Taken:**

* Premchand Lingamgunta – Design and Implementation of the registration page (2 days)
* Goutham Gandreddi – (5 days)