# **SOCIAL WORLD**

## PROJECT TITLE: SOCIAL MEDIA WEBSITE

## WEBSITE NAME: SOCIAL WORLD

## **TEAM MEMBERS:**

## RAVI TEJA TALARI – RXT170012

## AKHIL BITRA – AXB170061

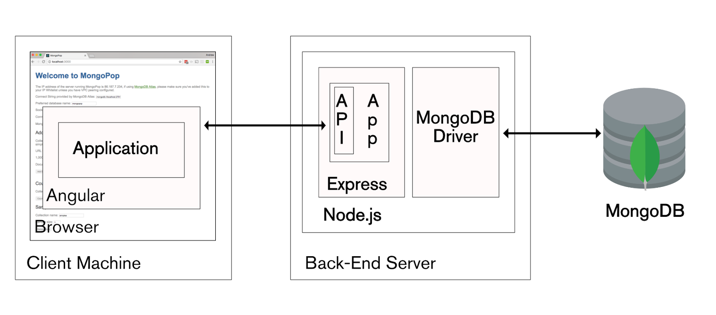
## VENKATA SAI PRANEETH VARMA KEERTHIPATI – VXK180026

## SRI JIGNASH REDDY ATTURU – SXA180079

# **SOCIAL WORLD**

## ARCHITECTURE:

The “Social World” is a web-based application based on MVC architecture. The architecture showing the entire flow of the web application is shown in the picture below.



## **DESCRIPTION:**

The following are the functionalities of our “Social World” website:

1. New User Registration:

A new user can register into the Social World website by clicking on the SignUp button that is available on the top right corner of the homepage. The new user must then enter his full name, email, password, confirm password and profile pic and then click on Sign Up button which is at the bottom of that page. If the user enters all the details without any flaws, he/she will be registered into the database and then redirected to the login page.

Rules for new user to register:

i) **email** field is unique which indicates that the user must give a new email id that is not previously registered.

Email is checked whether it is previously registered in the database by using AJAX.

ii) **password** should have at least eight characters ( at least one small letter, at least capital letter, at least one number, at least one special character)

iii) **confirm password** should be the same as the previously entered password

iv) **profile pic** is mandatory

* If all the above rules are followed, then the user will be directed to the login page which indicates that registration is successful.
* If any of the above rules are not followed, error messages will be displayed for that specific rule.
* Mandatory fields are checked whether they are filled or not by using JQuery
* Password rules are checked by using JQuery
* Passwords are hashed and stored in the database.

1. Existing user login:

The existing user ( already registered) must enter his email and password by clicking the login button on the homepage. If he enters the wrong email or password error message of “Invalid login credentials” will be displayed. If he enters the correct username and password, he will be logged into the website.

1. Website Landing page:

This page is the landing/home page as soon as the user logins with the correct information. This page consists of ‘social world’ logo, search bar where we can search for friends, and buttons like myprofile, privacy, message, notifications, logout whose functionalities will be explained below.

In addition, the user can post on this page. Once posted the post appears on this page.

1. “Post”/“Upload pictures” on landing page:

When the user is in the landing/home page, he/she can post. The user can write text, insert photo or video, disable comments to yes or no and then click on post button to get the post posted on his/her profile page.

1. “Search” for user:

When the user is in the home page, he/she can search for friends and view their profiles. When the user enters a name in the search box, he/she can see all the related names and then click on the respective names to view their profiles.

1. “Privacy”:

In the home page, there is privacy button. When the user clicks on this button, the user can see four different options: public, private, friends, and friends of friends.

* Public:

By default, the privacy setting will be public. Other users can see our profile.

* Private:

If privacy setting is selected as ‘private’ no one can see our profile other than admin. The admin is the only one who can see our profile.

* Friends:

If privacy setting is selected as ‘Friends’ only our friends can view our profile.

* Friends of Friends:

If privacy setting is selected as ‘Friends of Friends’ only our friends and their friends can view our profile.

1. MyProfile page:

When the user is logged into the website, he/she can view their profile by clicking on their name or photo (which is next to the name). MyProfile page is named with the specific user’s name and their photo.

When the user clicks on his/her name/photo he/she will be directed to his/her profile page. The user can post, view previous posts and can even search for other users from this page.

1. Friends profile page:

When the user enters the name of his friend in the search bar, the user can see all the related names and when he clicks on the view profile, he can see all the friends posts and comment on their posts.

1. Commenting on friend’s posts:

When the user goes into the friend’s profile, he can see other user’s posts. The user can then comment on other user’s posts by writing text and uploading images.

* If the friend opted for “Disable Comments - Yes” then the user cannot comment on that post. He should click on the “Request permission to comment” to ask his friend to give permission to comment on that post.
* If the friend opted for “Disable Comments - No”, the user can comment on that post. There is no need to take permission to comment on that post.

1. Messaging:

The user can chat with other users by clicking on the messaging tab. When the messaging button is clicked, the user will be directed to messaging page.

* He can see all his friends list on the left part of the page. He can click on that friend to start chatting.
* Alternatively, he can type in the friend’s name in the ‘friends search box’ to get the all the related names and then can proceed to chatting.
* Once the user clicks on ‘chat’, he can start chatting. The users friend profile picture will be displayed, and chat window will be opened at the center of the page.
* When the friend also logs into the system, example@example.com joined the chat will be displayed.

Example: george@gmail.com joined the chat.

* The user can then chat with his friends.

1. Search for other user’s posts:

When the user visits friends’ profile there is an input field where he can enter text to match with the texts on the posts. If there is a match, those posts will be returned.

1. Admin access:

The admin has the exact same interface as that of other users but have some special privileges like:

i) Accessing User Account – The admin can access all of users posts even if the profile is private

ii) Updating User Account – The admin can update privacy of users like make a user account public, private, friends or friends of friends for which there are special buttons added.

iii) Deleting User Account – The admin can delete any user account by clicking on the ‘delete account’ for that account

iv) Deleting specific content of User Account – The admin can delete any users’ posts.

The admin will be logging with email address of ‘admin@domain.com’

1. Sending a friend request:

The user should enter his friend name in the search box present in the home page. Then a list of all users will be displayed. The user should then click on his friend to go his profile where he can send friend request via ‘Add friend’.

His friend then gets friend request. He can either ‘accept’/ ‘reject’ the friend request.

## **LANGUAGES/ FRAMEWORKS USED:**

* FRONT-END: Angular 7 CLI, Bootstrap 4.3.1, JQuery 3.4.0, AJAX
* SERVER-SIDE PROGRAMMING: Node.JS 10.15.3, Express JS 4.16.4
* DATABASE: MongoDB 4.0
* WEB SERVICES: REST
* IDE: JetBrains WebStorm 2019, Atom, Studio3T

Initially, we thought of working with Java and JavaScript but later we found out that there are many additional features for MongoDB (dynamic database). Hence, we shifted to MongoDB.

* Studio 3T is used to view MongoDB and all the data stored in database
* The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications. You can use the tool directly in a command shell, or indirectly through an interactive UI such as Angular Console.

In addition to this, we used components for each of the module of our page. For example, we created a component each for registration, login, homepage, privacy, messaging, notifications etc. Each component consists of html, css, ts files where the code for front-end is stored.

* Bootstrap is an open source toolkit for developing with HTML, CSS, and JS.

We used Bootstrap for styling.

* jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. We used jQuery for form validation.
* Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. We used Node.JS for middleware.
* We used “multar” for handling images and videos in the back-end.
* We used “socket.io” for messaging purpose.
* Express.js, or simply Express, is a web application framework for Node.js

We used Express.js for routing purpose.

* MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata. We used MongoDB to store data.

## **DATABASE DESIGN:**

We used MongoDB database and Studio 3T is the IDE for MongoDB

After installing Studio 3T do the following:

i) Open command prompt and type mongod

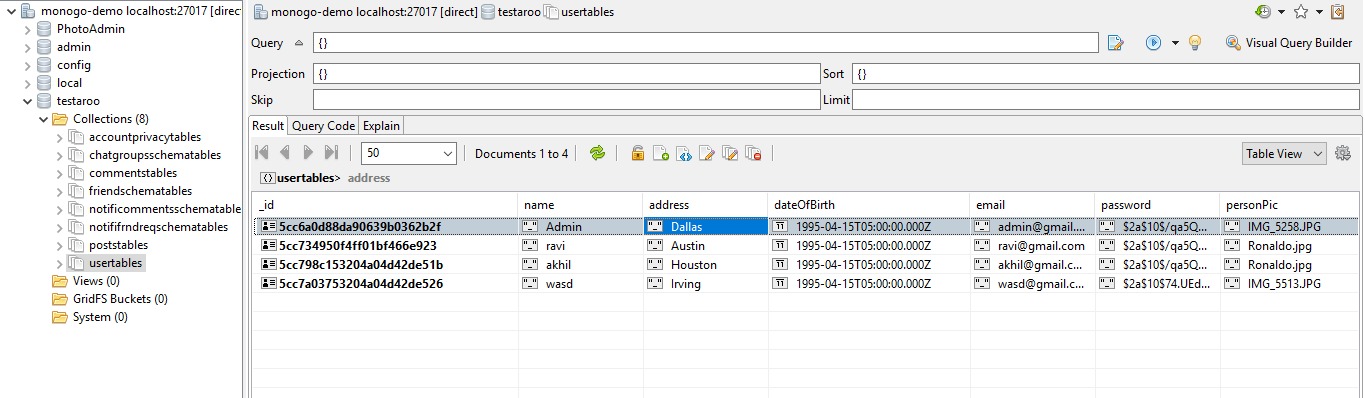
ii) Open another command prompt and type mongo

Then open Studio 3T and connect to mongodb by making a new connection to localhost:27017

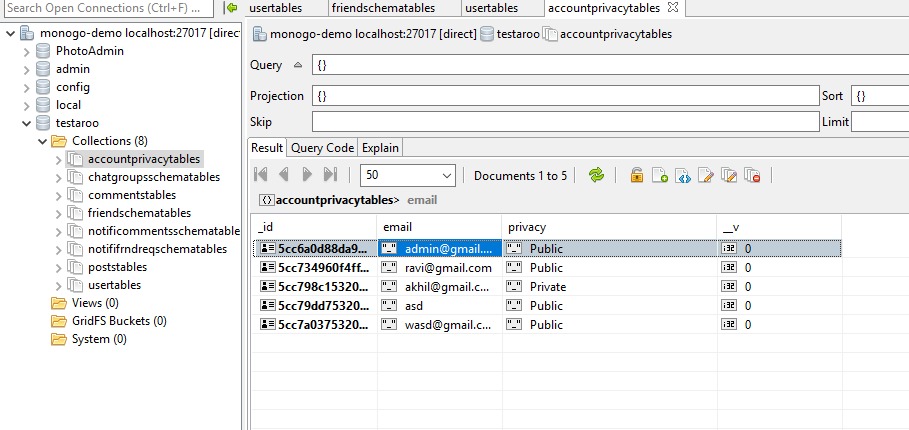
* Our database name is ‘testaroo’ and we created several collections like ‘usertables’, ‘accountprivacytables’,’poststables’, ‘commentstables’ etc...
* When a user registers into the website, an account will be created, and a document will be created into the ‘usertables’ collection.

Usertables consists of the fields like id, name, address, dateOfBirth, email, password, personPic, \_v

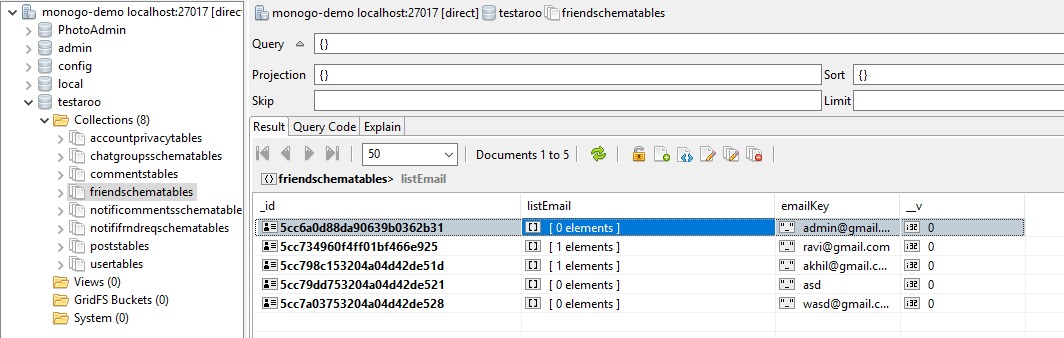
Each document will have a separate id.



* Depending on the privacy settings, user with email will have privacy.



* ‘friendschematables’ collection stores information about friends list, email, unique id



There are other collections like:

* Chatgroupsschematables
* Commentstables
* Notificationscommentsschema
* Posttables
* Notificationsfrndrequests

## **WORK DIVISION:**

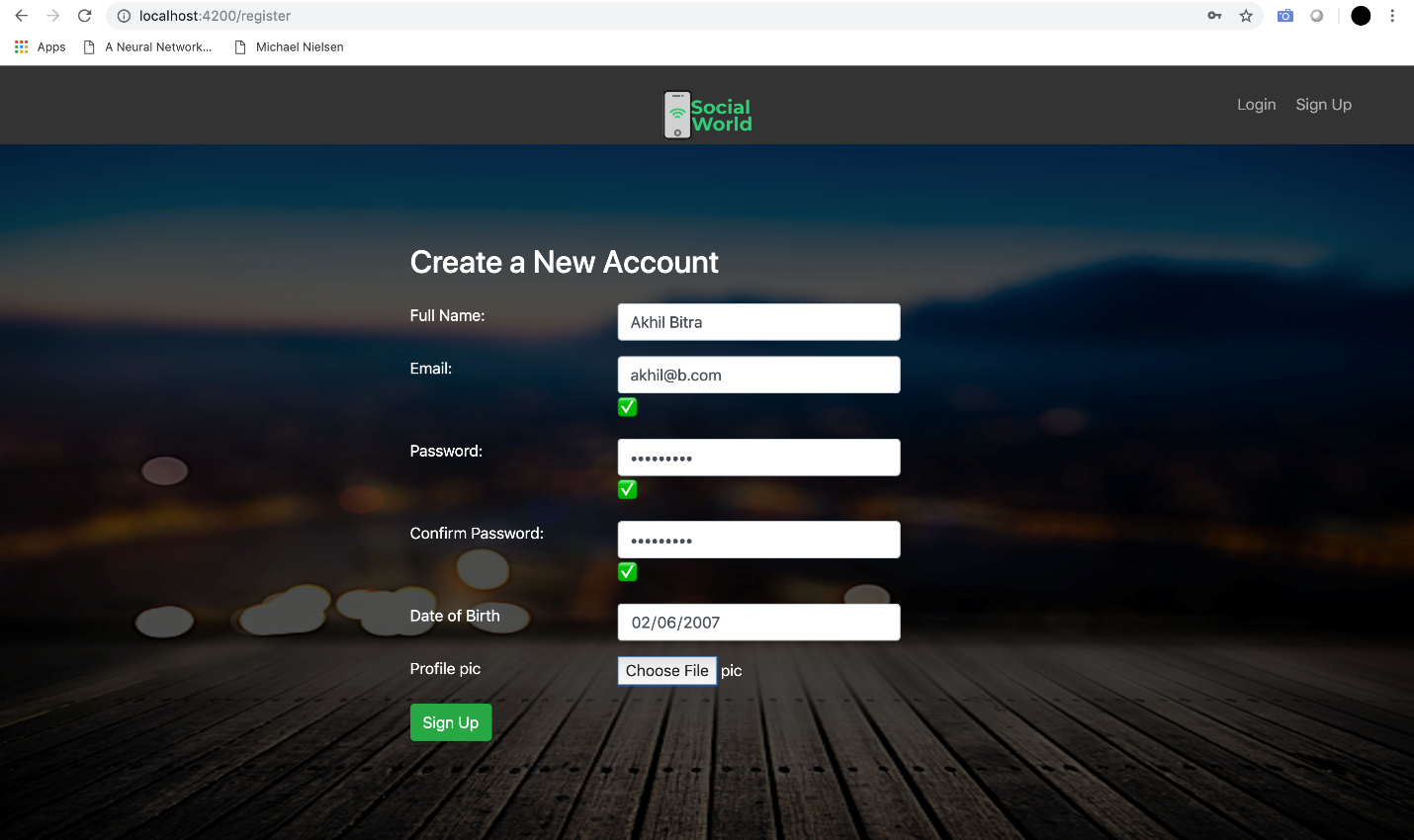
* Akhil Bitra did registration for a new user, login for an existing user which are described on page number 3 and posting on home page described on page 4
* Praneeth did posts, search for user and privacy described on page 4
* Jignash did user’s profile, friend’s profile, commenting on friend’s posts described on page 5
* Ravi Teja Talari did messaging, admin, handling of collections in mongodb which are described on pages 5,6,7

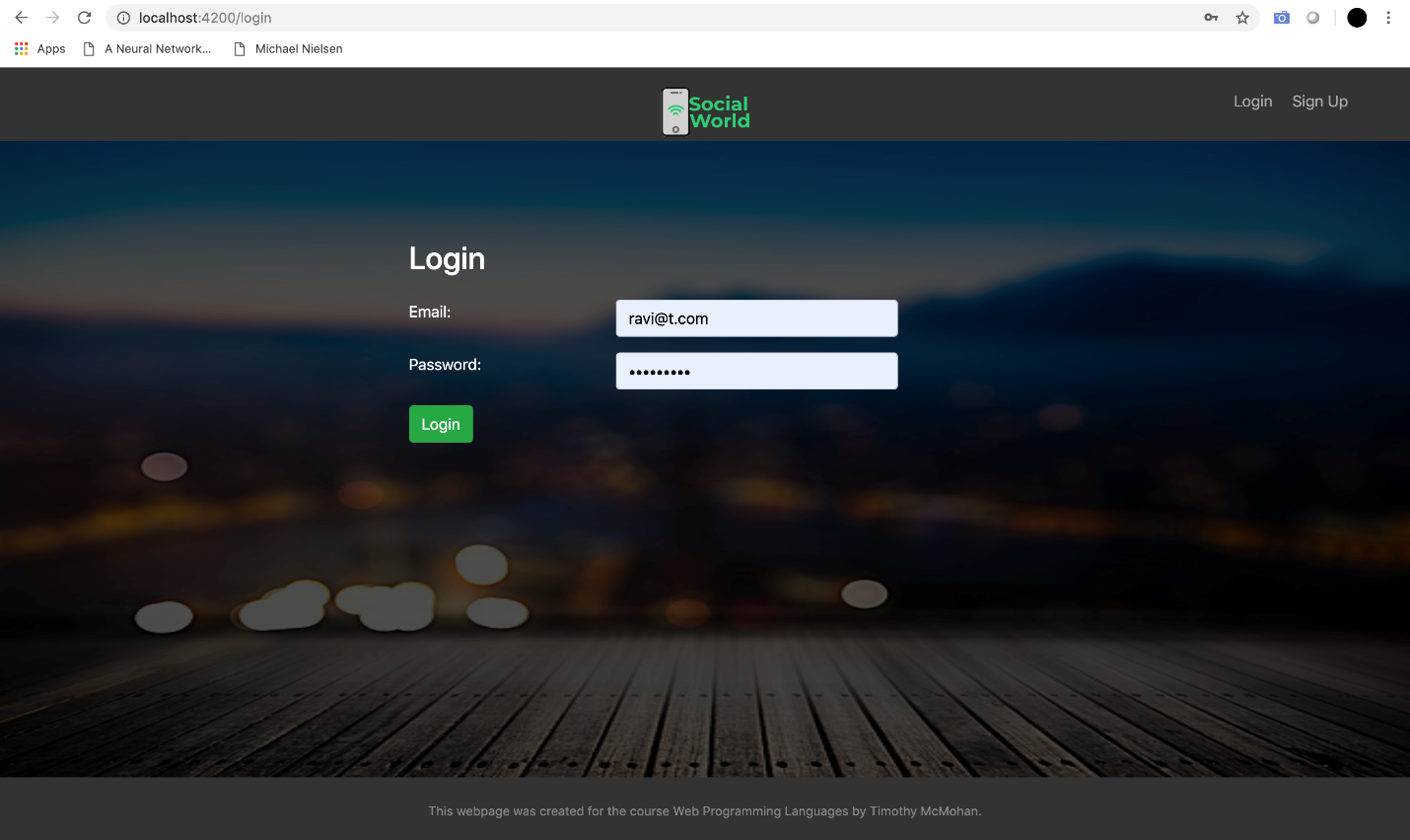
## **UNIQUE FEATURE:**

* We added a new component named ‘Connect!’ in our website. This button when clicked shows one random user who are registered in the database.
* When we navigate out to another page and come back again to this page, another new user will be shown.
* So, the unique feature is finding random users and viewing their profile based on their privacy settings.

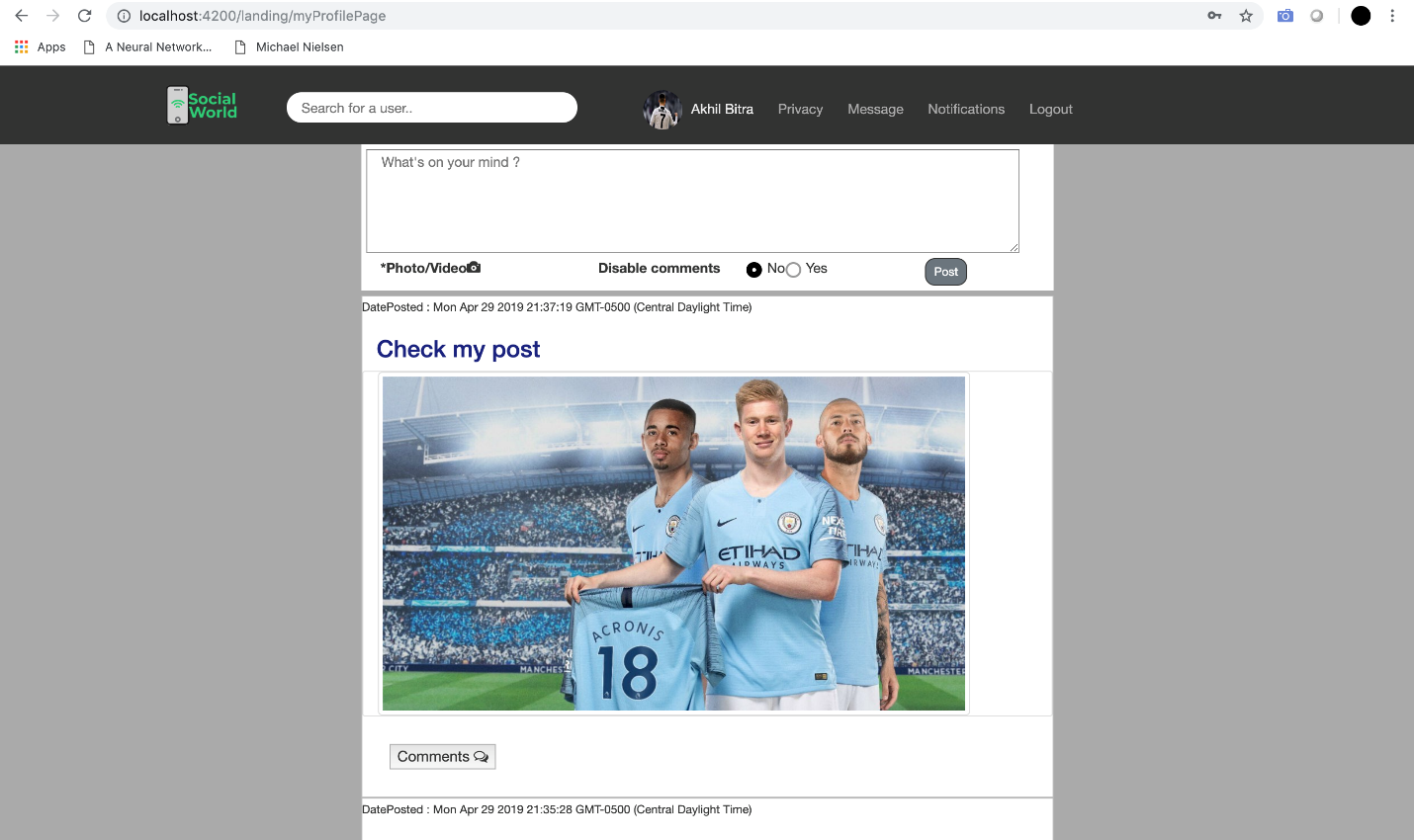
## **SCREENSHOTS:**

1. Registration page:

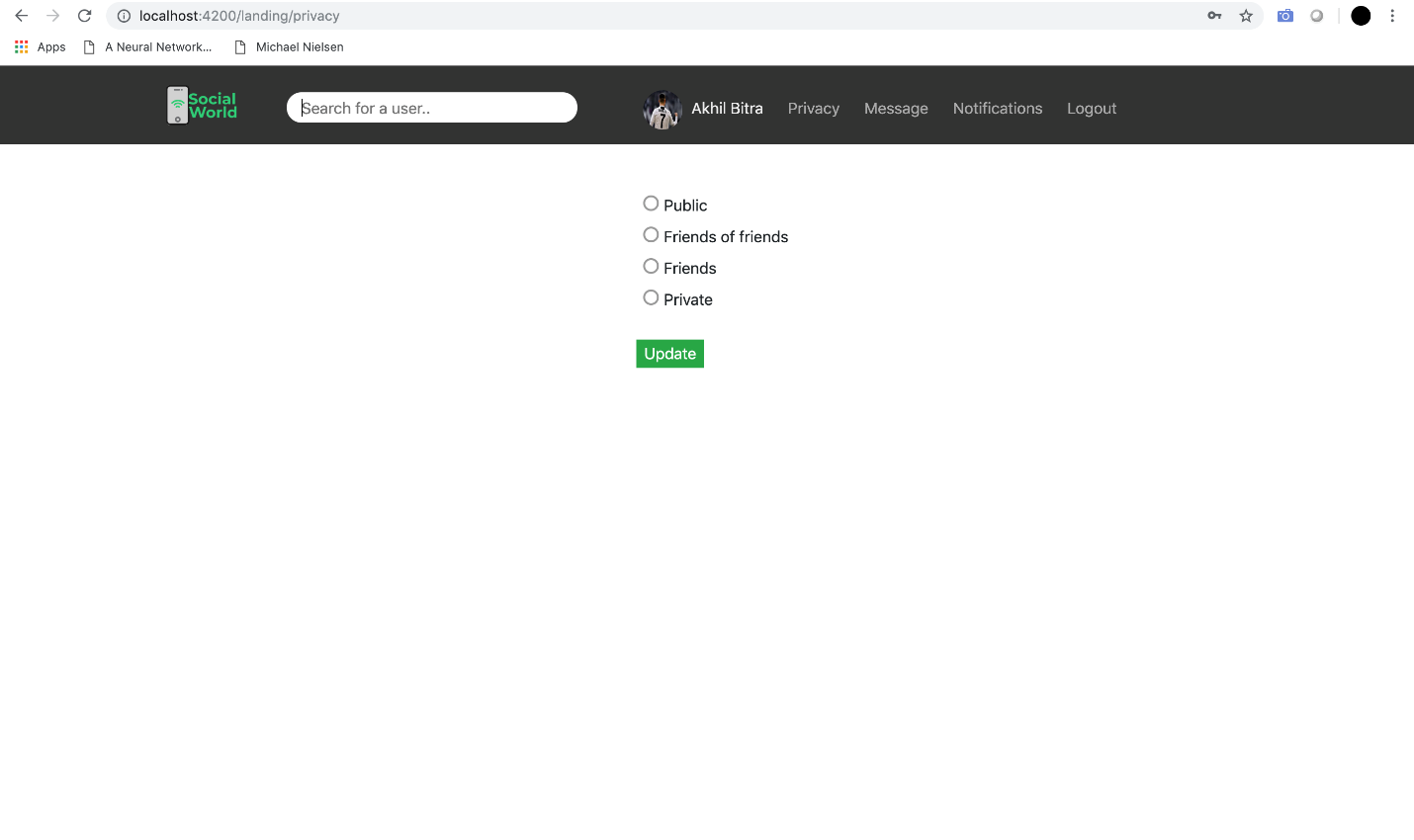


2. Login Page:

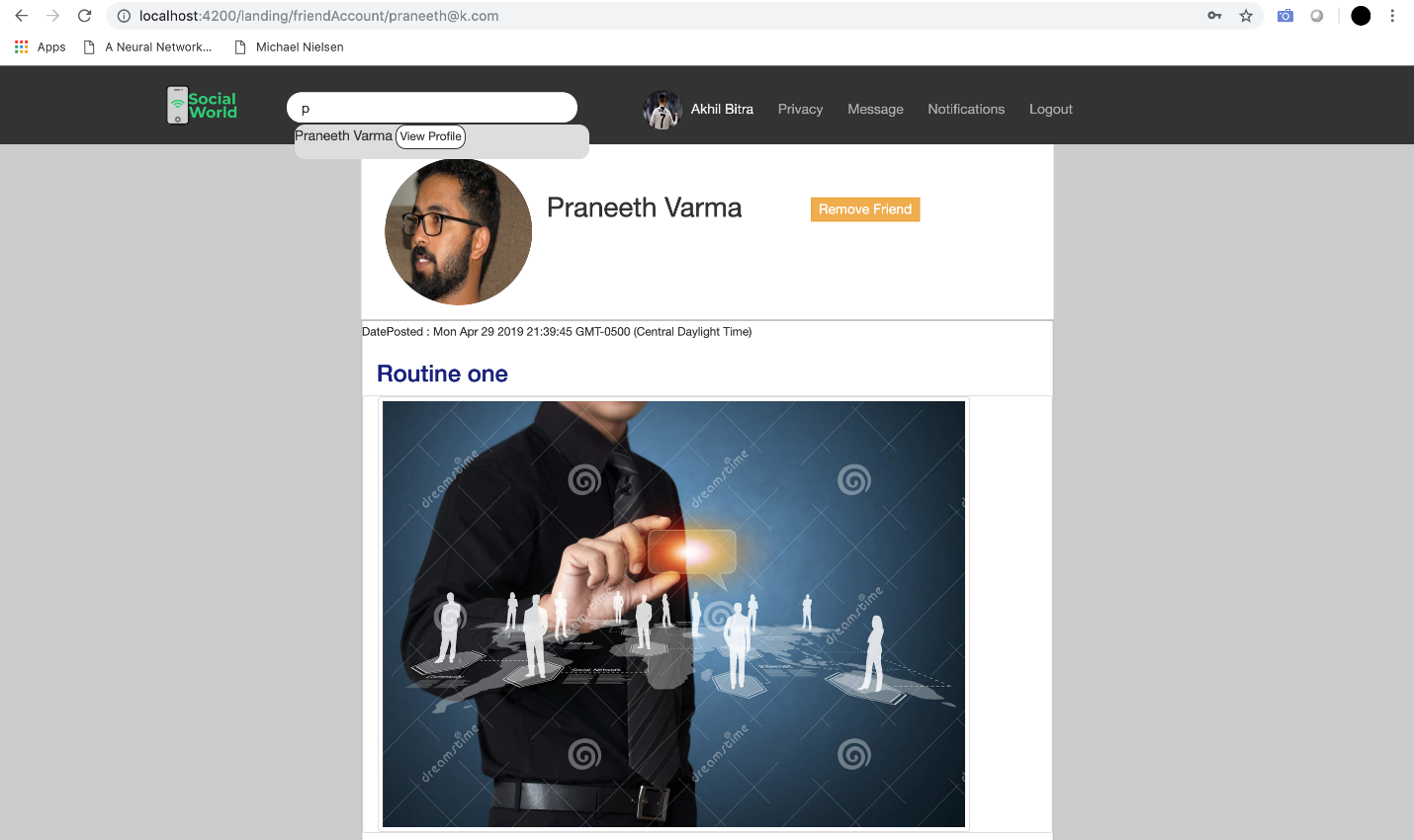
3. User profile:



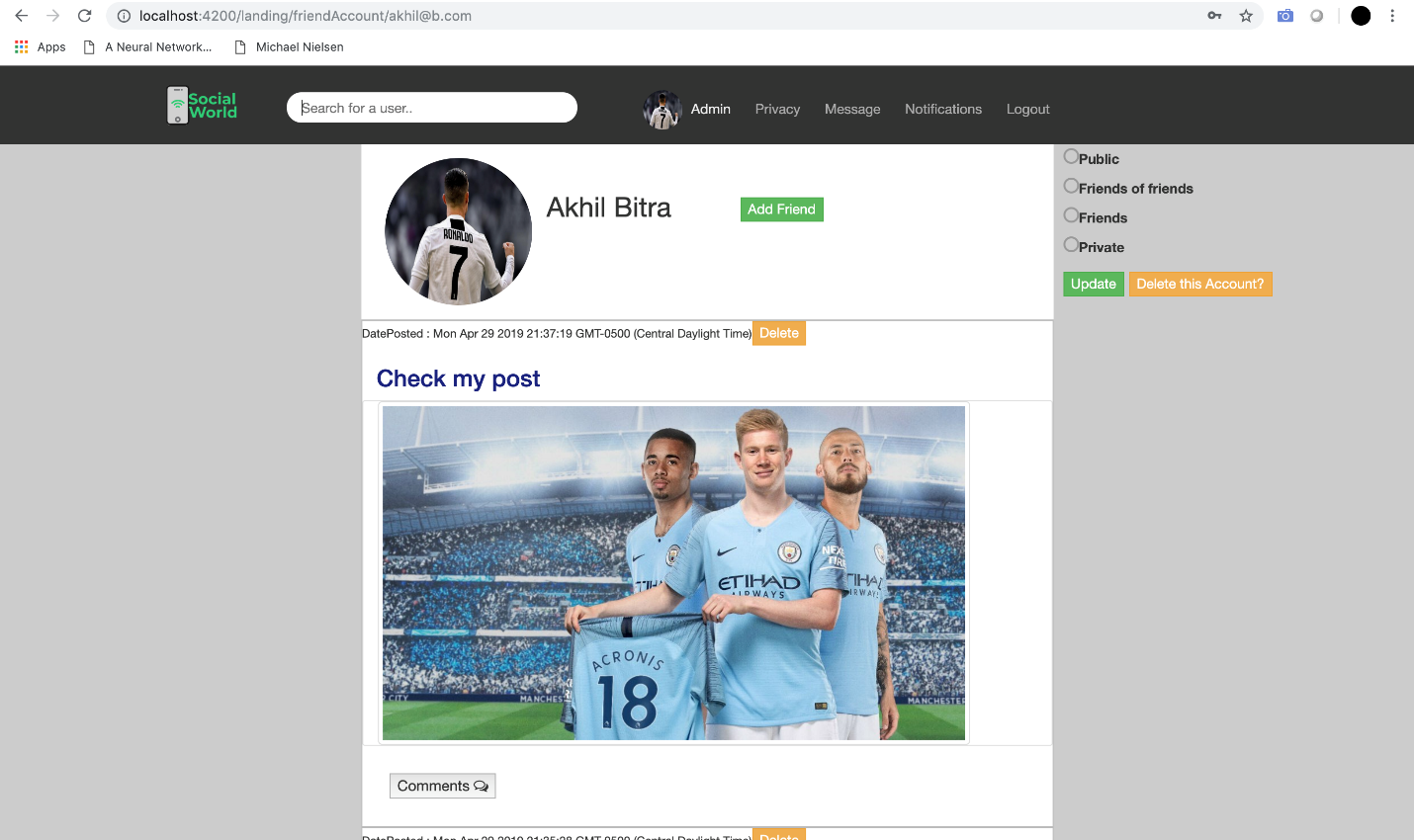
4. Privacy settings



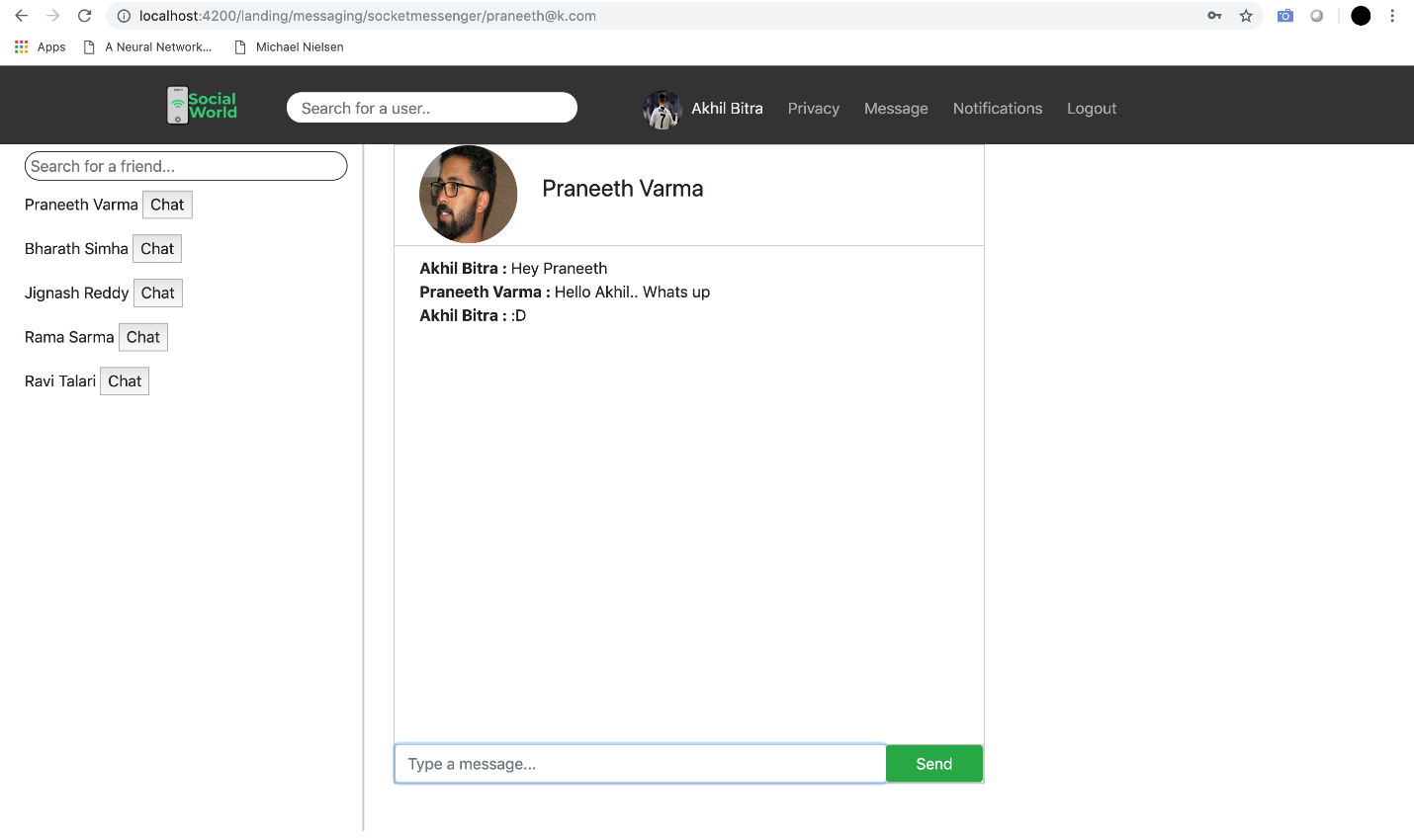
5. Friends page:



6. Admin profile:



7. Messaging:



**HOW TO RUN THE APPLICATION:**

1. Connect to the mongo server by entering mongod in one command prompt and entering mongo in another command prompt.

2. Run the back-end server by giving the below command in the folder containing back-end code:

npm run start

3. Run the front-end angular by giving the below command in the folder containing front-end code

npm serve –open