**Chapter 1 Assignment Profile**

* 1. **Technology**
  2. **Setup**
  3. **Testing**
  4. **Technology:**

Front-End: HTML5, CSS3, Bootstrap, Angular9

Backend: NodeJS, MongoDB

* 1. **Setup:**

I have used latest version of UI framework angular and it requires angular CLI to run the project. Moreover, I have implemented backend logic as an API. Therefore, there will be two projects; one for UI and another for backend. The project setup details are as follow.

**UI Project:**

Name: CardGame-UI

Dependencies: Angular CLI, node

Commands:

1. **npm install** **–g @angular/cli** (Note: Install angular CLI)
2. **npm install** (Note: Go inside project directory and type this command; it will install required modules for project)
3. **ng serve --port 8081** (Note: This command will run project in localhost on port 8081, port specification is required because I have used **CORS** policy in backend api. So, it will not allow any request to be processed from another port except 8081)

**Backend Project:**

Name: CardGame-Api

Dependencies: node, MongoDb

Commands:

1. **npm install** (Note: Go inside project directory and type this command; it will install required modules for project)
2. **node server.js** (Note: This command will start the server on port 3000)

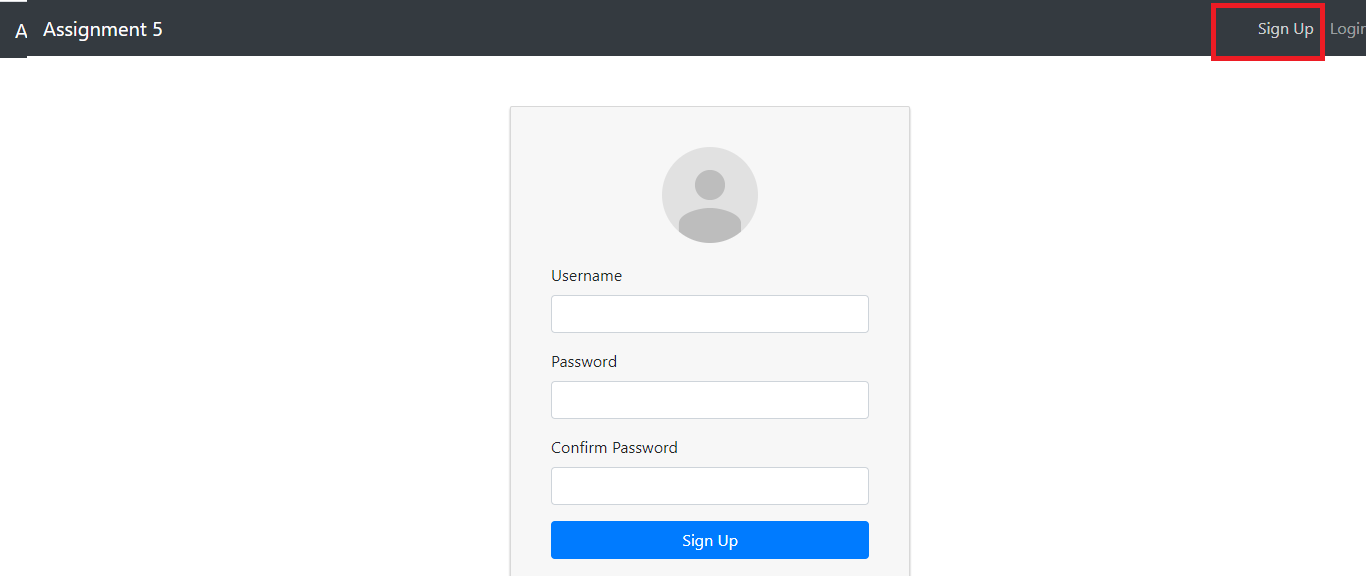
\***Note:**

Both projects are needed to be run in order to test the implemented functionalities as mentioned in the project documentation.

* 1. **Testing:**

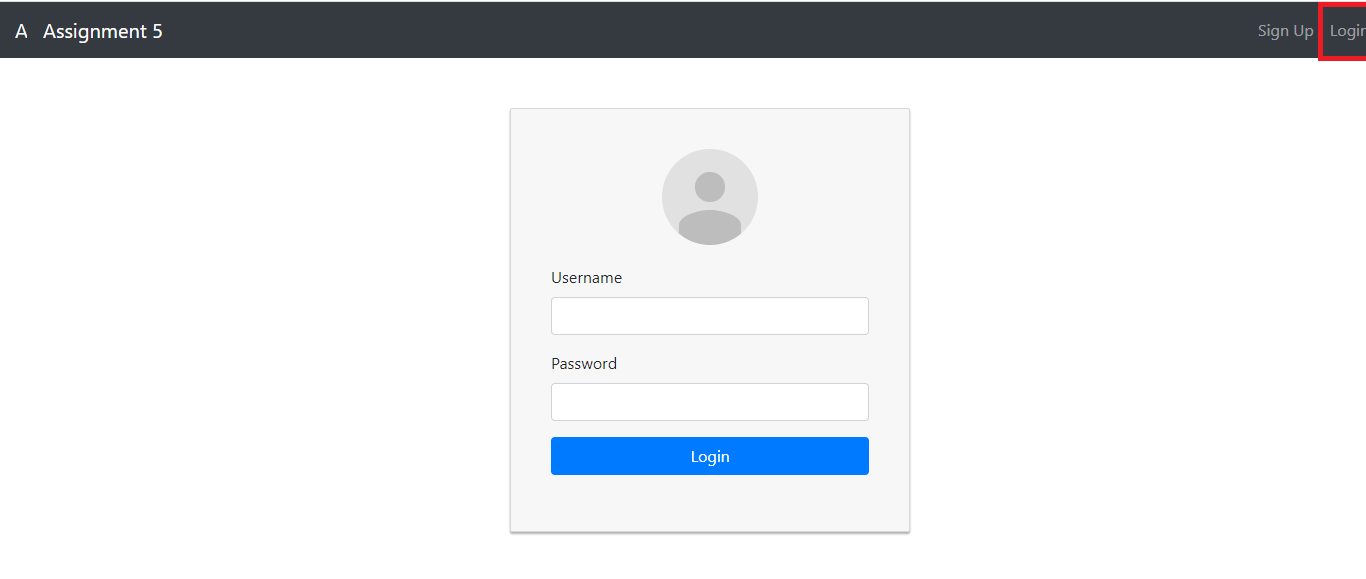
First of all, for the attractive and easy interaction with UI; I have bifurcated assignment functionalities in three pages **Home, People and Trade**.

Initially, users have to register on portal from signup form and user can go to that page by clicking on **SignUp** menu available in the top right corner of header.



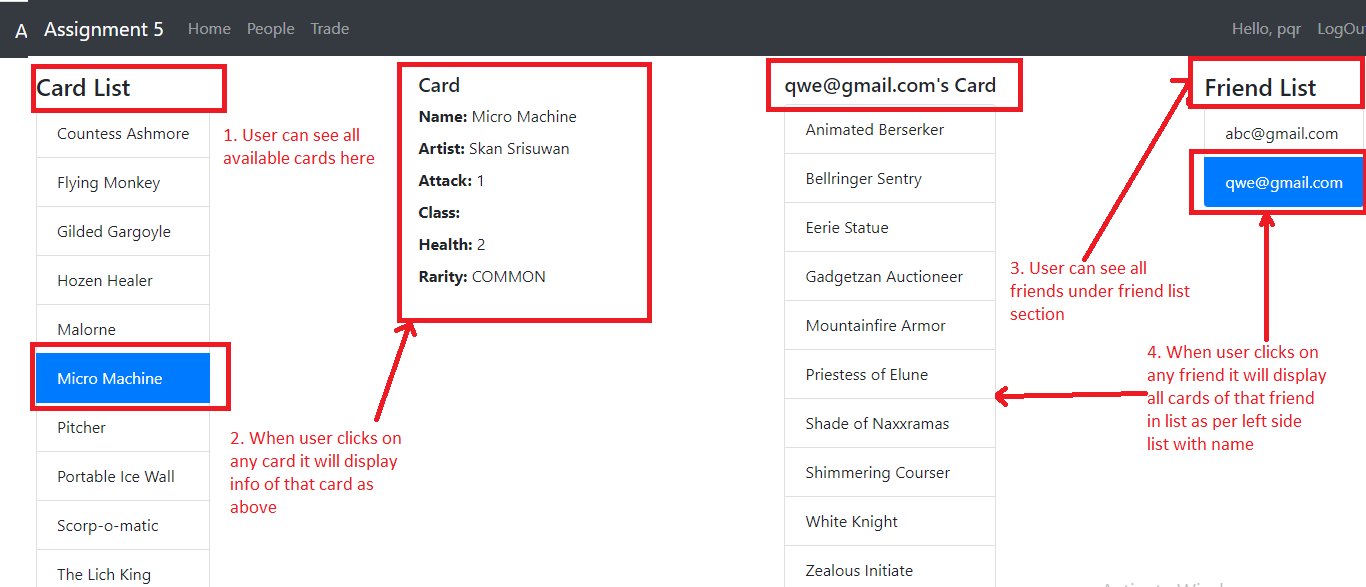
* 1. **Registration Page**

Next, on successful registration user will be redirected to **LogIn** screen with success message, users have to enter their credentials and they will be navigated to **Home** page if username and password are right, otherwise they will receive error message in top right corner of the browser.



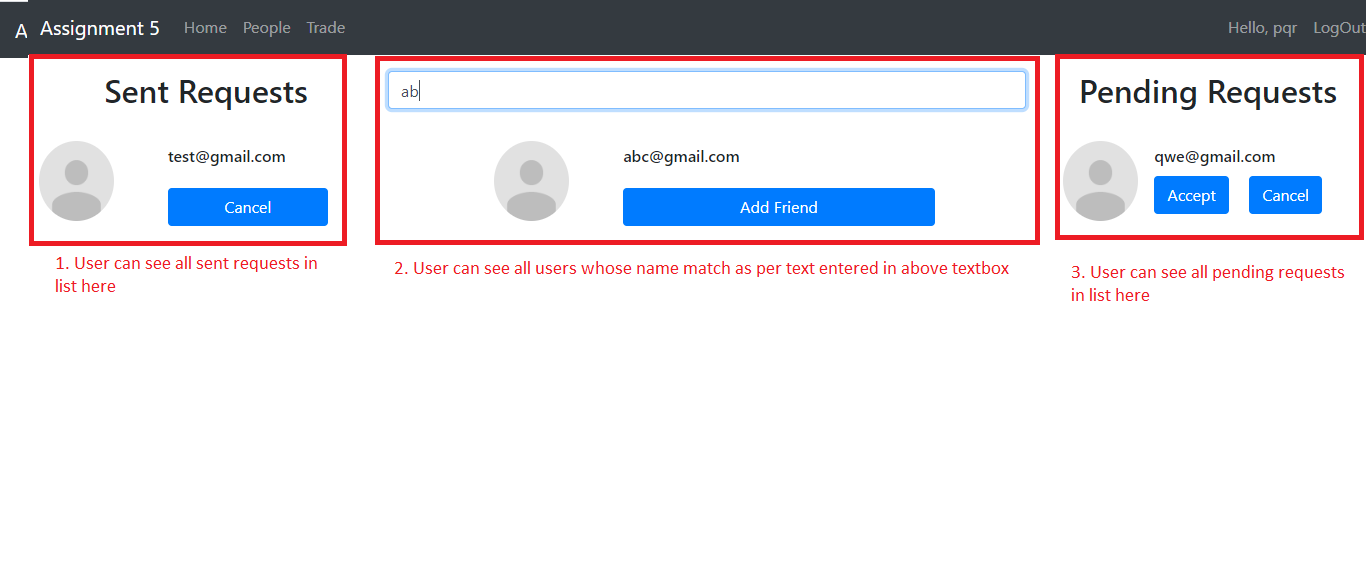
* 1. **Login Page**

At **Home** page, users can see their basic profile like available cards and those cards info as well as all their friend cards.



**1.3 Home Page**

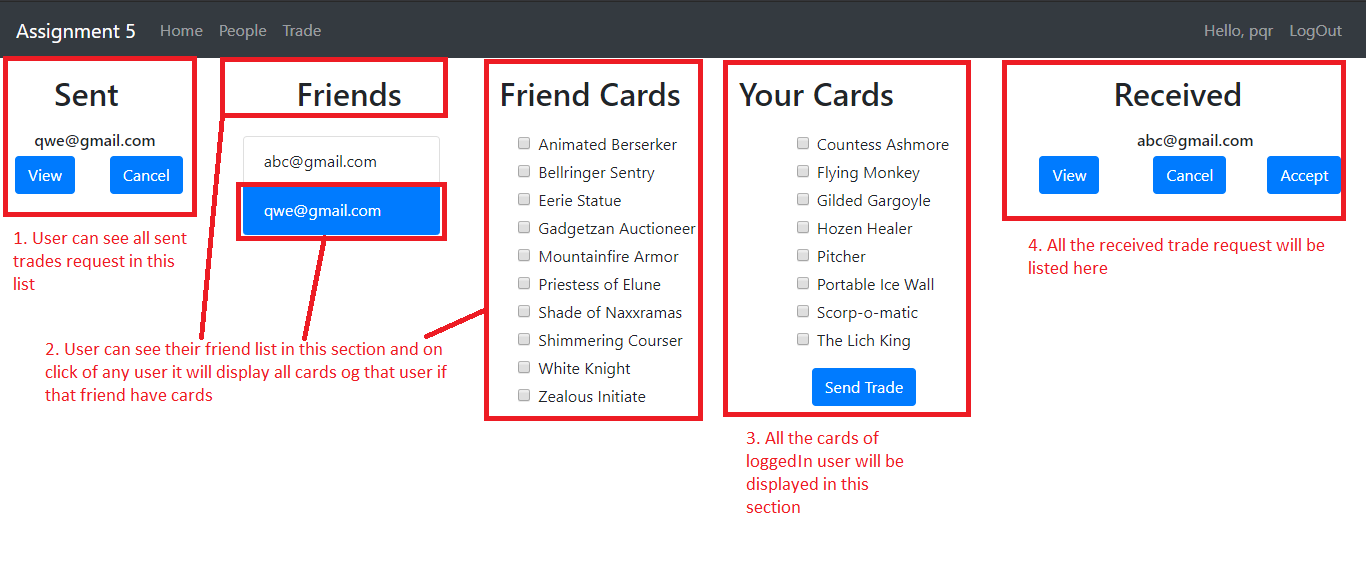
On the **People** window, users can do operations like search friend, accept or reject friend request and they can see sent requests and pending requests. In addition, user cannot send request to other people if they have already sent or received from those people and they can’t send request to people, who are already their friends and they will see message in top right corner after each action.



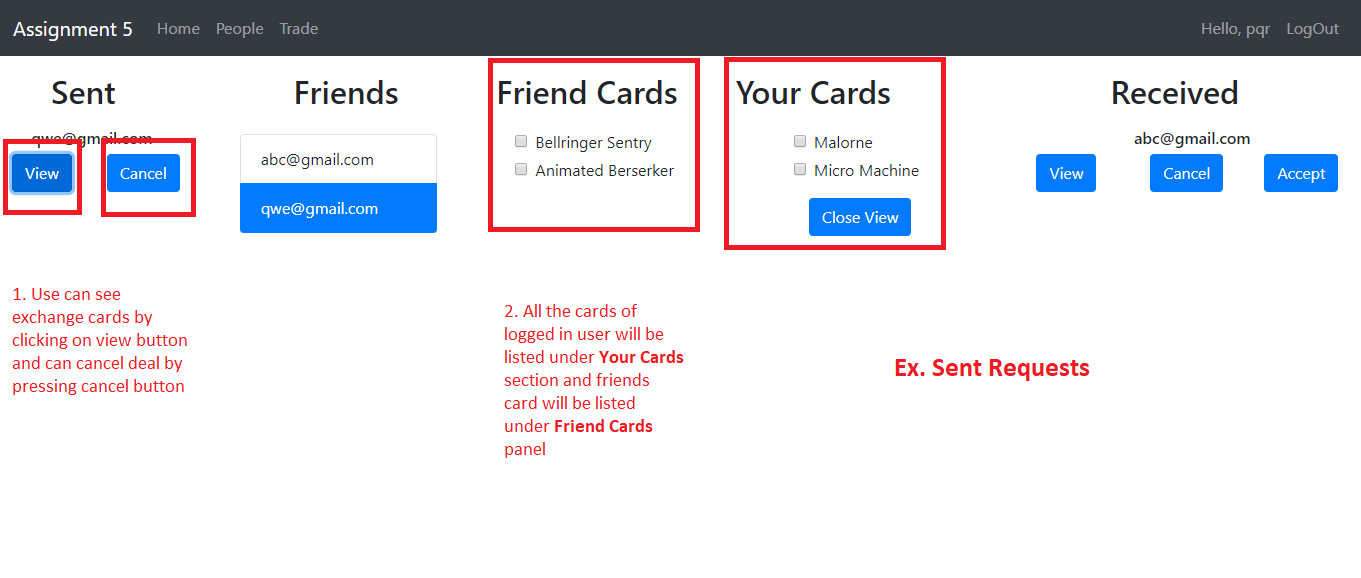
**1.4 People Page**

At **Trade** screen, user can see all the sent and received requests and also they can see all friends with their cards. Furthermore, user can send request using **Send Trade** button and they are able to view, cancel or accept the trade deal using available options.

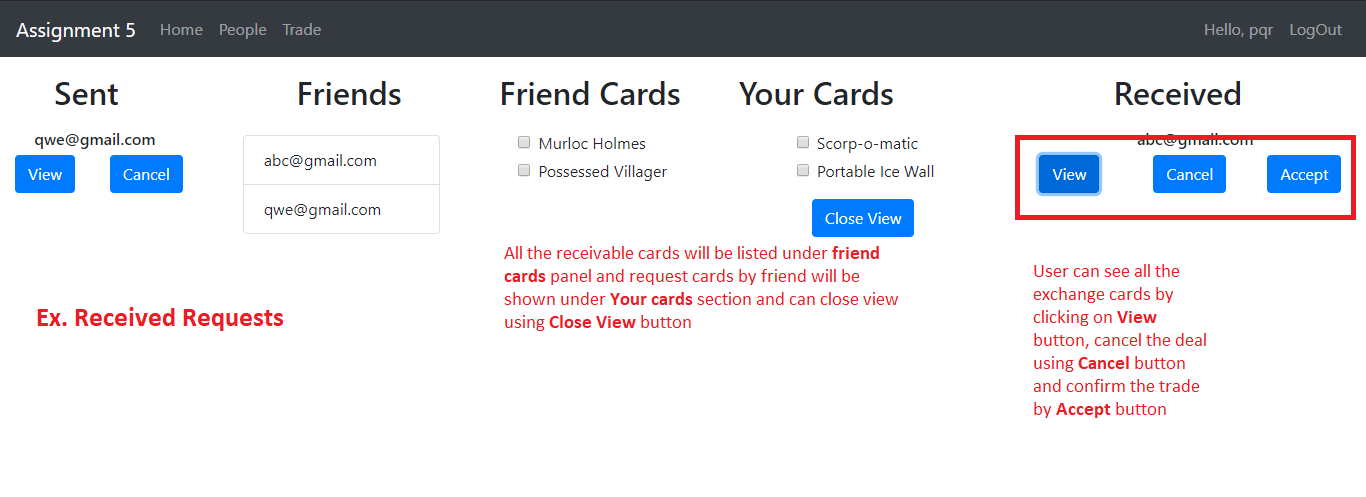
**\*Note:** For the elimination of repeated card trading, I’ve used approach of remove the cards of user when they send trade request. Therefore, they will not be able to send trade request more than once for one card.



**1.5 Trade Page (Default View)**



**1.6 Trade Page (Sent Requests View)**



**1.7 Trade Page (Received Requests View)**

**\*Note:** All the data will be updated real-time as specified in document.

**Chapter 2 System Profile**

* 1. **Design Decisions**
  2. **Data Storage**
  3. **Route Detail**

**2.1 Design Decisions:**