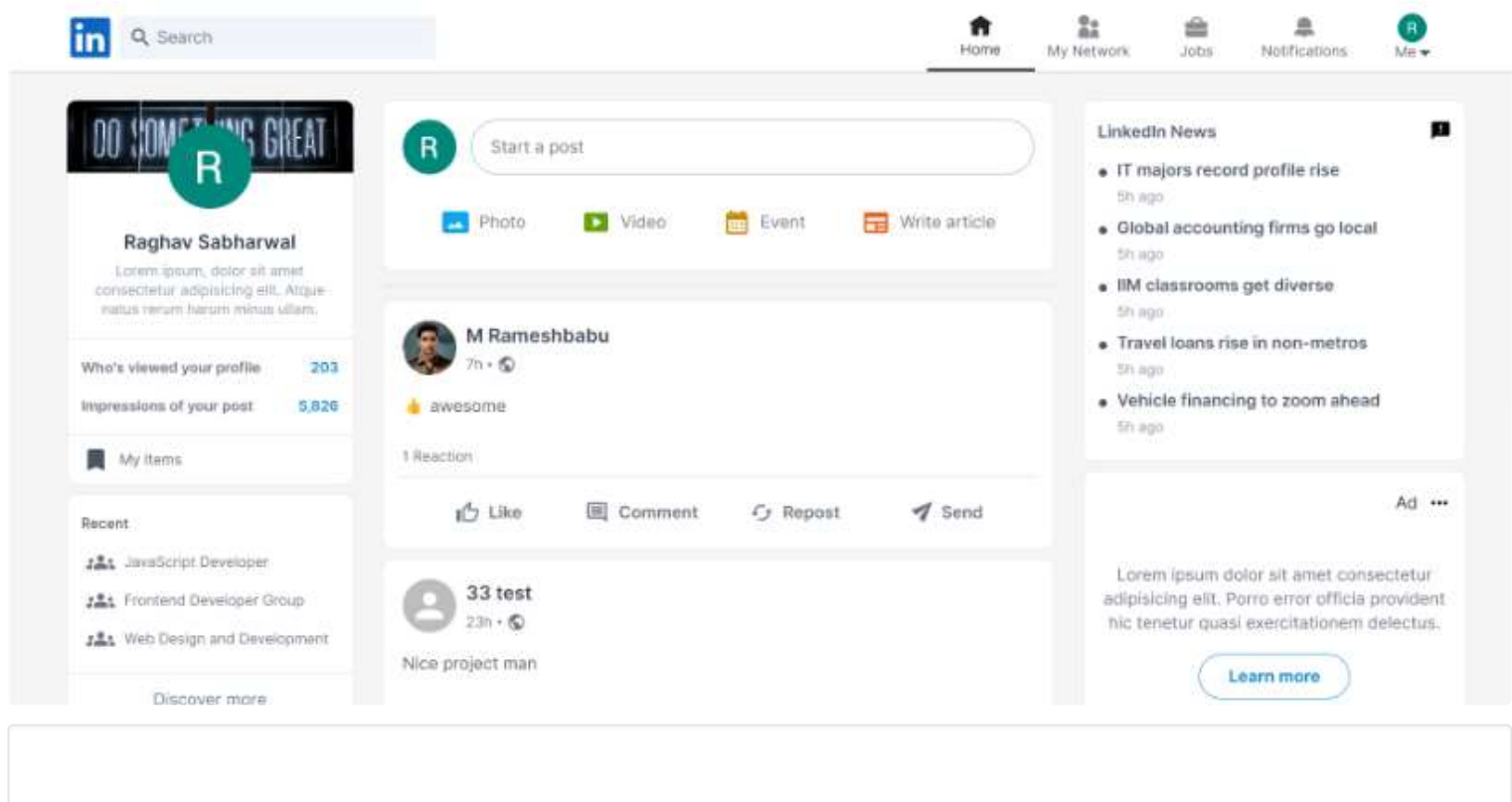


## Project Objective

👉 The project aims to create a **LinkedIn clone** (<https://in.linkedin.com/>) application using HTML, CSS, JavaScript, and React. The app will replicate the essential features of LinkedIn, view posts, experience of other users, add post, interact with the post, share experiences, post updates, and search for job opportunities.



## Project Context

👉 LinkedIn is a popular professional networking platform used for job searching, professional development, and building professional connections. The LinkedIn clone will offer users a similar experience, enabling them to create professional profiles, connect with others, and explore job opportunities.

***Tech Stack Prerequisite:*** HTML, CSS, JavaScript, React

## Note on Protected Routes

👉 Auth related APIs are protected (update user data, update password), So to use them, follow the instruction properly.

👉 Remember, for any protected routes, you'll need to include the JWT token in the header:

```
headers: {  
  'Authorization': 'Bearer YOUR_JWT_TOKEN',  
  'projectId': 'YOUR_PROJECT_ID'  
}
```

👉 To know, what is JWT and how JWT works? Watch: [What Is JWT and Why Should You Use JWT](#)

👉 How to Use JWT Authorization using Postman? Watch: [How to Use JWT Authorization](#)

👉 Every API request must include a 'projectId' in the headers. If it doesn't, the request will be rejected with an error.

--->Project ID here is the hash of the Playground URL


--->Example: <https://my.newtonschool.co/playground/project/f104bi07c490>

--->Your Corresponding Project ID would be: **f104bi07c490**

---> Note: These projectId will be unique for each question.

Kindly refer to the link given below for implementing API:

<https://drive.google.com/file/d/1K-TiaalOWFYuMd-L3wtHJAKVgW7xyeNI/view>

 [my.newtonschool.co/playground/project/8nbih316dv01](https://my.newtonschool.co/playground/project/8nbih316dv01)

## Project Steps

### Step 01: Set up the project and UI design

- 👉 Create a new React project using **the templates given**.
- 👉 Organize the project directory structure, including subdirectories for components, styles, and other necessary files.
- 👉 Install any required dependencies, such as **react-router** for navigation.
- 👉 The UI of this project requires huge amount of CSS to make the application look alike the real one.
- 👉 It is better to use external CSS libraries like **Material UI** (<https://mui.com/material-ui/>) which is a popular React component library that follows Google's Material Design guidelines. It provides a set of ready-made components for various UI elements which can help us to create complex UI in easier way by using the predefined CSS classes.
- 👉 The UI of the application must be responsive in nature.

**For further queries, please go through the link below:**

<https://learn.newtonschool.co/n/9pxkmxt9d6kw/preview>

## Step 02: Navbar

👉 The website should have a navigation section.

👉 The navigation bar should have a **logo of the website**, which should point toward the home page of your website.

👉 Organize the app section by categories for easy navigation and better user experience. Like **Home, Networks, Jobs, Messaging, Notification**, etc.

👉 Add a **Search Bar**. This will allow users to easily search for text matching with posts, communities, comments, and users.

👉 There should be an User icon with the user's name, once clicked the button, it should display **Login/ Logout, Try Premium, Profile** and **Dark Mode Toggle** Options as a dropdown.

--> If the user is not logged in point toward the Login page once clicked on the **"Login"**.

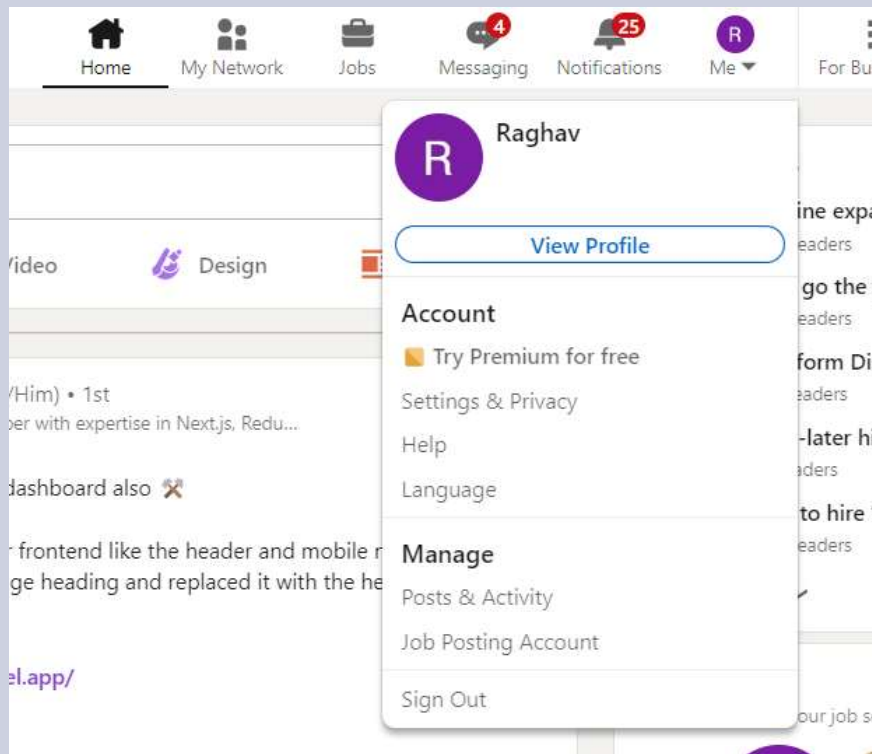
--> If the user is logged in, point towards the Logout page once clicked on the **"Logout"**

--> Once clicked on the **Try Premium**, it redirect to the Subscription page details are mentioned below.

--> Once clicked on the **Profile** option, it should redirect the user to user's profile page, showing details, like, username, first name, last name and emails, etc.

--> Allow user to toggle between **light and dark mode** of the app for a more comfortable viewing experience.





## Step 03: Homepage

- 👉 The Home page consists of various sections, each in a grid-based format to provide uniformity to the data showing on the screen and tackle it in an easier manner.
- 👉 It contains a Navbar on top that contains many functionalities given in the next section.
- 👉 Below the Navbar, there is a create post option available that you can also access from the Navbar.
- 👉 Below the create post option, we have the posts which show the post uploaded by a user in reverse chronological order (latest post on top).
- 👉 The posts are aligned in a vertical view, one below the other, and infinite scroll is implemented on them so more posts will render when the user will scroll down.
- 👉 **GET: Use the below APIs for fetching posts**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/post', {  
  headers: {
```

```
'projectId': 'YOUR_PROJECT_ID'
```

```
}
```

```
})
```

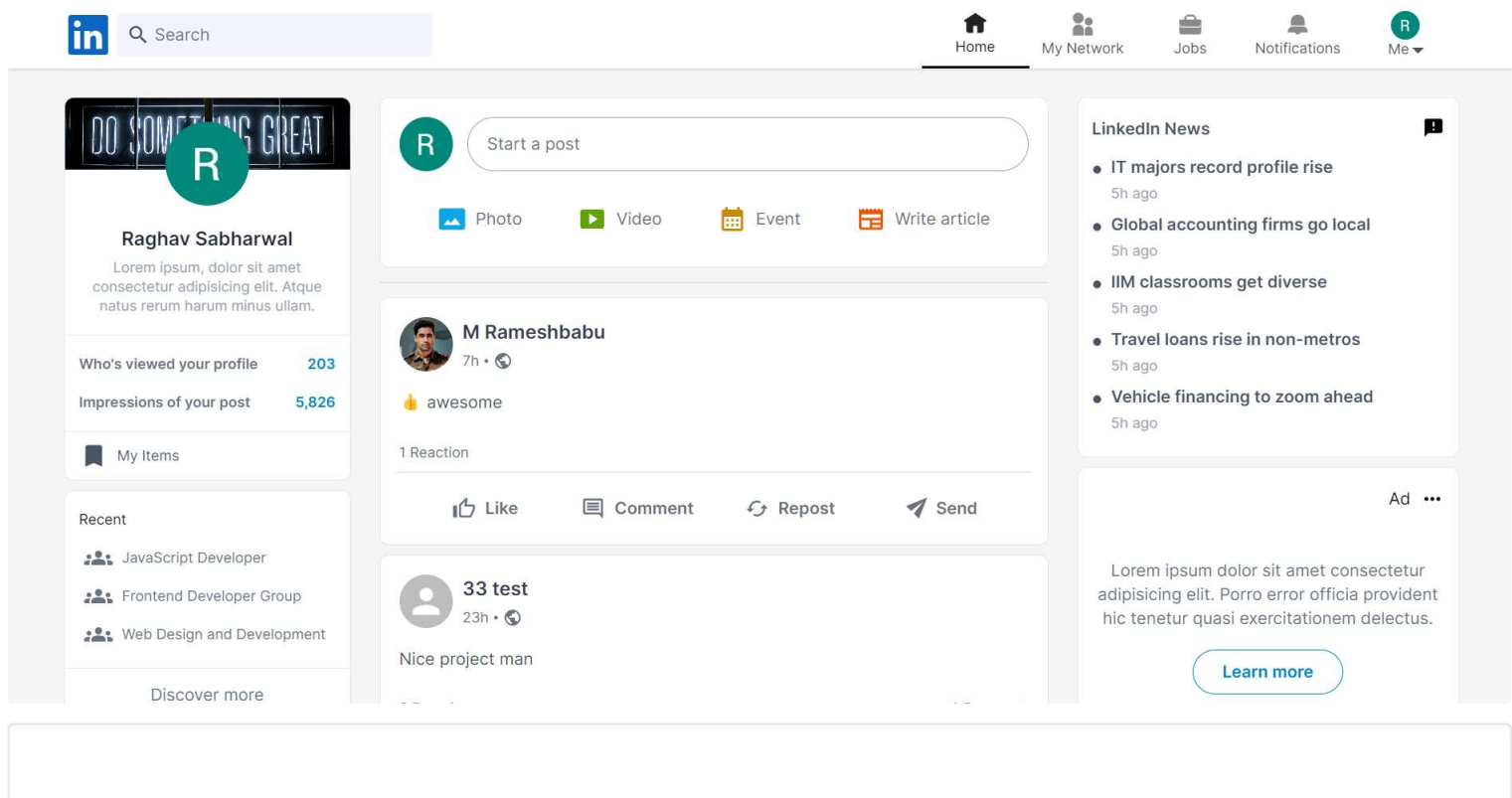
👉 The API will send a response of 20 posts, add the `?limit=100` param in the endpoints of the API to get 100 post. The resulting API will be

`https://academics.newtonschool.co/api/v1/{{app_name}}/post?limit=100` .

👉 Include post actions: **Like, Comment, and Share/ Send.**

### 👉 **Pagination or Infinite Scroll:**

Pagination can be handled with page and limit parameters. For instance, GET `/api/v1/linkedin/post?limit=10&page=2` would get the second page of results, with 10 results per page.



## Step 04: Post Component

👉 Design a reusable post component to display each post's content, timestamp, author, and interaction options (like, comment, share, etc).

👉 **POST: Use the below API for Upvoting a Post**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/like/:postId', {  
  headers: {  
    'Authorization': 'Bearer YOUR_JWT_TOKEN',  
    'projectId': 'YOUR_PROJECT_ID'  
  }  
})
```

👉 **GET: Use the below API for fetching comments of a Post**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/post/:postId/comments', {  
  headers: {  
    'Authorization': 'Bearer YOUR_JWT_TOKEN',  
    'projectId': 'YOUR_PROJECT_ID'  
  }  
})
```

👉 **POST: Use the below API for creating comment of a Post**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/comment/:postId', {  
  headers: {  
    'Authorization': 'Bearer YOUR_JWT_TOKEN',  
    'projectId': 'YOUR_PROJECT_ID'  
  },  
  body: {  
    'content': 'YOUR_COMMENT'  
  }  
})
```

👉 **PATCH: Use the below API for updating comment of a Post**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/comment/:commentId', {  
  headers: {  
    'Authorization': 'Bearer YOUR_JWT_TOKEN',  
    'projectId': 'YOUR_PROJECT_ID'
```

```
    },  
    body: {  
      'content': 'YOUR_UPDATED_COMMENT'  
    }  
  })  
}
```

### 👉 **PATCH: Use the below API for updating comment of a Post**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/comment/:commentId', {  
  headers: {  
    'Authorization': 'Bearer YOUR_JWT_TOKEN',  
    'projectId': 'YOUR_PROJECT_ID'  
  },  
  body: {  
    'content': 'YOUR_UPDATED_COMMENT'  
  }  
})
```

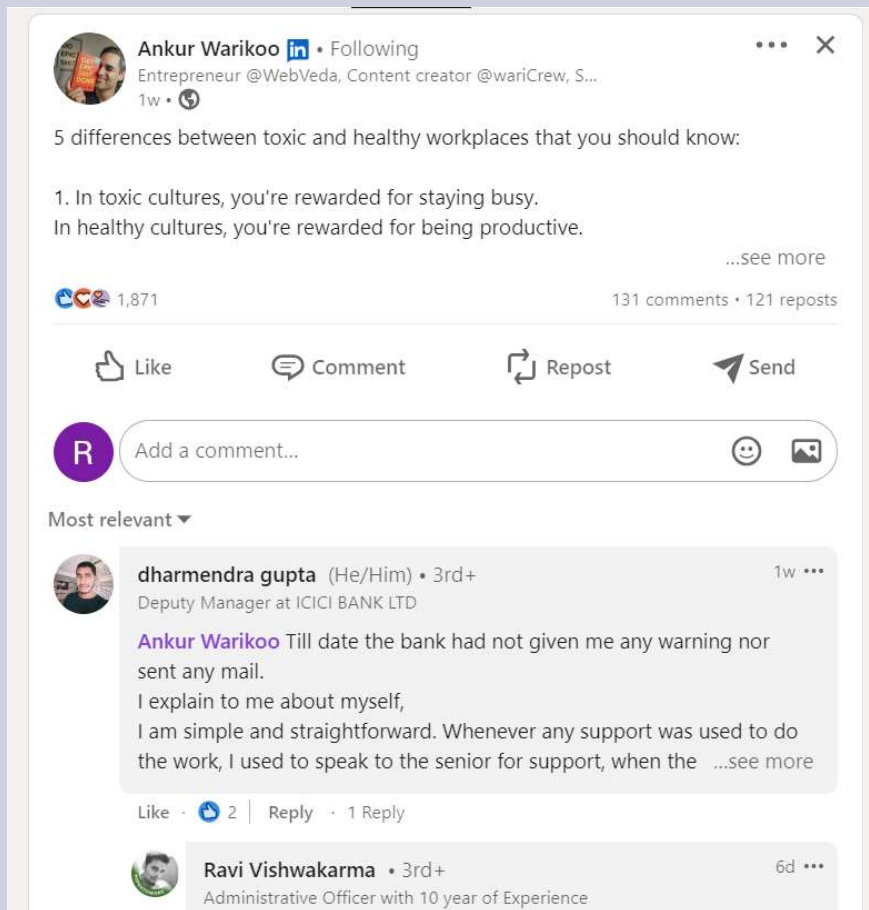
👉 Show the comment section as well below the post.

👉 Display the total number of likes and comments for each post.

👉 Enable users to comment on posts and engage in discussions within specific post.

👉 Allow users to reply to comments for threaded conversations.





## Step 05: Search Functionality

👉 Implement a search bar to allow users to search for job postings based on keywords, location, and industry.

👉 Display search results in real-time as users type their queries.

👉 You can use the below search API or the filter API for searching:

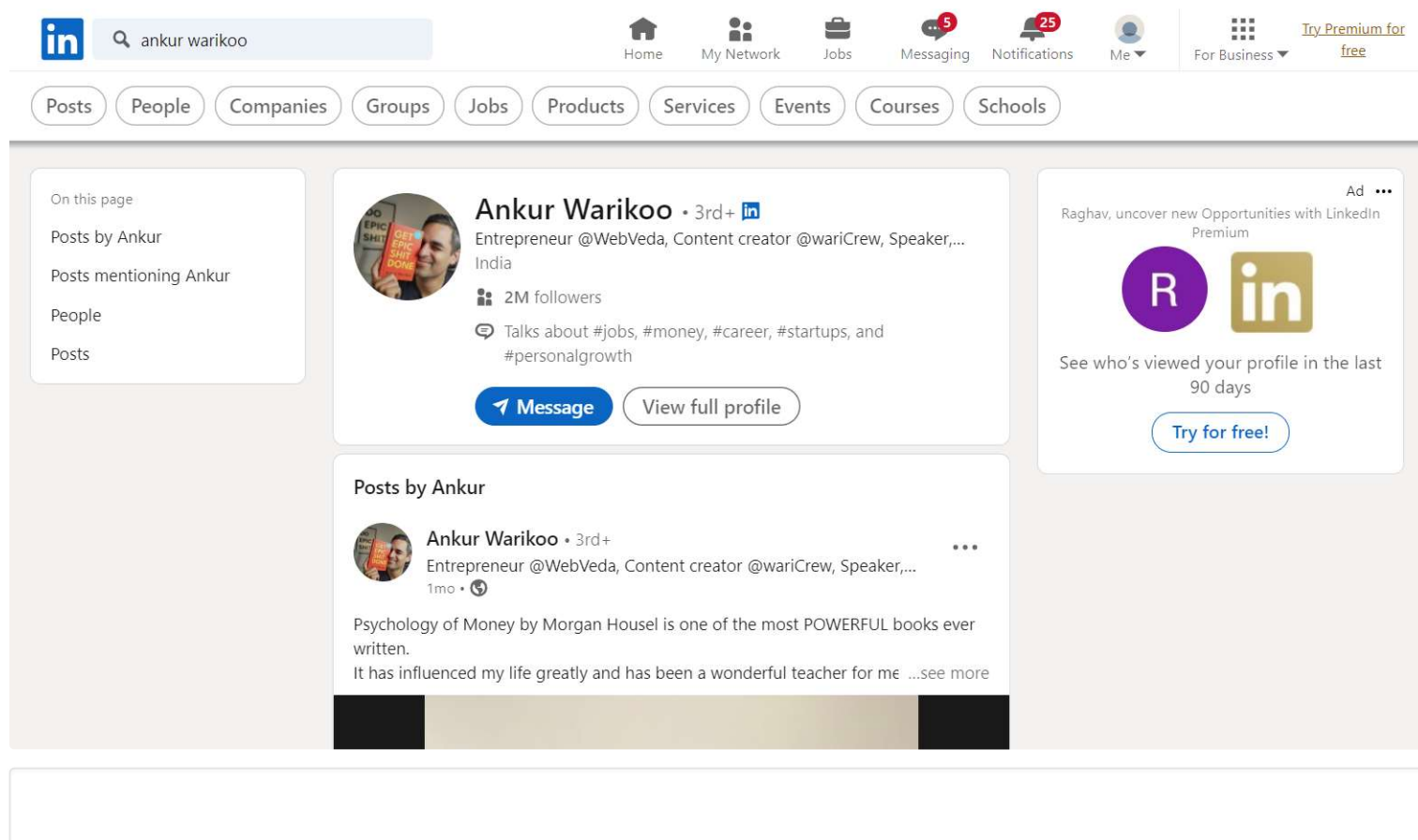
```
fetch('https://academics.newtonschool.co/api/v1/linkedin/post?search=
{"field":"search_term_here"}', {
  headers: {
    'projectID': 'YOUR_PROJECT_ID'
  }
})
```

```

fetch('https://academics.newtonschool.co/api/v1/linkedin/post?filter=
{"field":"search_term_here"}', {  headers: {
  'projectId': 'YOUR_PROJECT_ID'
}
})

```

Here, "field" will your key you want to search post by. It can be the **title** of the post, **author** of the post, or **content** of the post. Ans "search\_term\_here" would be the users' search input.



## Step 06: Login / Signup page

👉 Before the User can create posts or view other's profile, check if the user is registered and login into the application. Since we are working with the Frontend only, use the below routes for Registering the user, and then login the user. Ask the user for it's **first name**, **last name**, **email**, and **password**.

👉 If the user is not login, redirect the user toward the login page.

👉 Also, make sure users are logged in to access creating, liking a post, view others' profile. If a user attempts to like, display a modal prompting them to either log in if they have an account or register if they're new to the platform. This helps in encouraging user engagement and capturing potential new users.

👉 Add **reset/ update password** functionality.

### 👉 *Login:*

```
fetch('https://academics.newtonschool.co/api/v1/user/login', {  
  method: 'POST',  
  headers: {  
    'Content-Type': 'application/json',  
    'projectId': 'YOUR_PROJECT_ID'  
  },  
  body: JSON.stringify({  
    email: 'user_email',  
    password: 'user_password',  
    appType: 'linkedIn'  
  })  
})
```

### 👉 *Signup:*

```
fetch('https://academics.newtonschool.co/api/v1/user/signup', {  
  method: 'POST',  
  headers: {  
    'Content-Type': 'application/json',  
    'projectId': 'YOUR_PROJECT_ID'  
  },  
  body: JSON.stringify({  
    name: 'user_name',  
    email: 'user_email',  
    password: 'user_password',  
    appType: 'linkedIn'  
  })  
})
```

### 👉 Update Password:

```
fetch('https://academics.newtonschool.co/api/v1/user/updateMyPassword, {  
  method: 'PATCH',  
  headers: {  
    'Content-Type': 'application/json',  
    'projectId': 'YOUR_PROJECT_ID'  
  },  
  body: JSON.stringify({  
    name: 'user_name',  
    email: 'user_email',  
    passwordCurrent: 'current_password',  
    password: 'user_new_password',  
    appType: 'linkedIn'  
  })  
})
```



Make the most of your professional life

Agree & Join

OR

Already on LinkedIn? [Sign in](#)

## Step 07: Compose a Post

- 👉 A logged in user should be able to add a new post.
- 👉 A new post should have a **description and image (which can be optional)**.
- 👉 It majorly contains three things which are content, and the tags which you can attach to your tweet which determines the category they belong to .

### 👉 **POST: Use the below API for Creating a Post**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/post/', {
  headers: {
    'Authorization': 'Bearer YOUR_JWT_TOKEN',
    'projectID': 'YOUR_PROJECT_ID'
  },
  body: {
    'title': 'postTitle',
    'content': 'postContent',
    'images': 'postImage',
  }
})
```

### 👉 **PATCH: Use the below API for Updaing a Post**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/post/:postId', {  headers: {
    'Authorization': 'Bearer YOUR_JWT_TOKEN',
    'projectID': 'YOUR_PROJECT_ID'
  },
  body: {
    'title': 'new postTitle',
    'content': 'new postContent',
    'images': 'new postImage',
  }
})
```

### 👉 **DELETE: Use the below API for Deleting a Post**

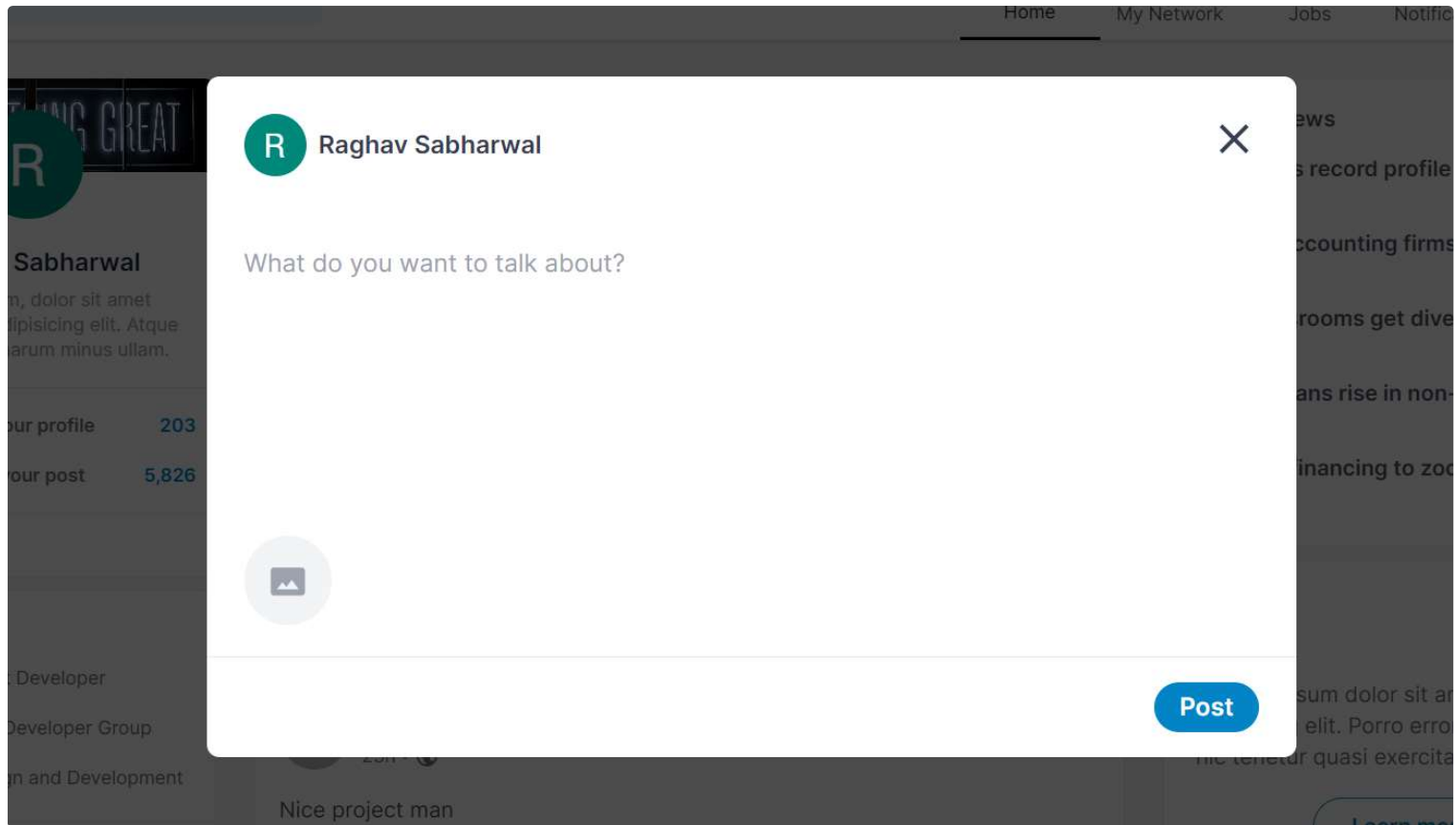
```
fetch('https://academics.newtonschool.co/api/v1/linkedin/post/:postId', {  headers: {
```

```
'Authorization': 'Bearer YOUR_JWT_TOKEN',
```

```
'projectId': 'YOUR_PROJECT_ID'
```

```
}}
```

👉 User should also be able to like existing posts on the home page.



## Step 08: User Profile and Networking/ View groups

👉 Create user profile pages showing **"name"**, **"address"**, **"phone"**, **"workExprience"**, **"education"**, **"about"**, **"skills"**, etc.

👉 Allow users to view other Profiles as well.

👉 **GET: Use the below API for getting user by ID**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/user/:userId', {
```

```
headers: {  
  'Authorization': 'Bearer YOUR_JWT_TOKEN',  
  'projectId': 'YOUR_PROJECT_ID'  
}  
})
```

👉 Allow user to follow and unfollow user profile. Use the below APIs for achieving this functionality.

👉 **POST: Use the below API for following an user by ID**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/follow/:userId', {  
  headers: {  
    'Authorization': 'Bearer YOUR_JWT_TOKEN',  
    'projectId': 'YOUR_PROJECT_ID'  
  }  
})
```

👉 **DELETE: Use the below API for unfollowing a Profile**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/follow/:userId', {  
  headers: {  
    'Authorization': 'Bearer YOUR_JWT_TOKEN',  
    'projectId': 'YOUR_PROJECT_ID'  
  }  
})
```

**NOTE: Follow/ Unfollow functionality should only be functional with following profiles and not with pages.**

👉 A logged in user should be able to create a new group.

👉 Allow users to view other view groups as well.

👉 Implement the ability for users to create new groups using the below APIs.

👉 **POST: Use the below API for Creating a Group**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/channel/', {  
  headers: {  
    'Authorization': 'Bearer YOUR_JWT_TOKEN',  
    'projectId': 'YOUR_PROJECT_ID'
```

```
    },
    body: {
      'title': 'postTitle',
      'description': 'postDescription',
      'images': 'postImage',
    }
  })
}
```

### 👉 **PATCH: Use the below API for Updating a Group**

```
fetch('https://academics.newtonschool.co/api/v1/linkedin/channel/:channelId', {
  headers: {
    'Authorization': 'Bearer YOUR_JWT_TOKEN',
    'projectID': 'YOUR_PROJECT_ID'
  },
  body: {
    'title': 'postTitle',
    'description': 'postDescription',
  }
})
```

**Note:** The above API only updates the title and description of the post and not the image. Provide only the title and description in the body.

### 👉 **DELETE: Use the below API for Deleting a Group**


```
fetch('https://academics.newtonschool.co/api/v1/linkedin/channel/:channelId', {
  headers: {
    'Authorization': 'Bearer YOUR_JWT_TOKEN',
    'projectID': 'YOUR_PROJECT_ID'
  }
})
```

### 👉 **GET: Use the below API for getting post of a particular Group**


```
fetch('https://academics.newtonschool.co/api/v1/linkedin/channel/:channelId/posts',
{
  headers: {
    'Authorization': 'Bearer YOUR_JWT_TOKEN',
    'projectID': 'YOUR_PROJECT_ID'
  }
})
```








[Home](#)[My Network](#)[Jobs](#)[Notifications](#)

Me



### Dhanush Prabhu S

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Magni natus, sed asperiores totam atque tenetur architecto facilis aliquid.

Connecting and Messaging feature is not available

[Connect](#)[Message](#)[More](#)


#### About

Lorem ipsum dolor sit amet consectetur adipisicing elit. Nesciunt perferendis quas natus possimus, earum fuga recusandae saepe voluptatem tempora libero!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Rem quibusdam voluptatem placeat eos illo dolore sequi eius, laudantium harum porro dignissimos veritatis repellendus. Deleniti neque sunt dolor harum consequatur! Numquam provident nostrum dolore molestias illum consequuntur eum fugit id laboriosam!

Ad

Raghav Sabharwal, boost your job search with Premium


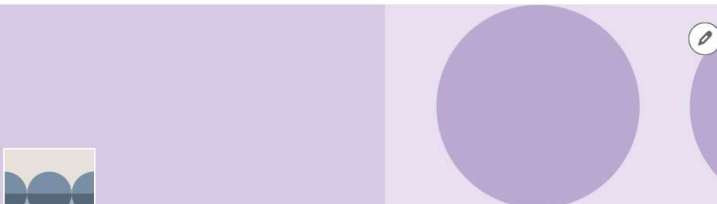


See who's viewed your profile in the last 90 days

[Try for free!](#)

Profile View

Create group



Group name\*

Inspiring Entrepreneurs in DC

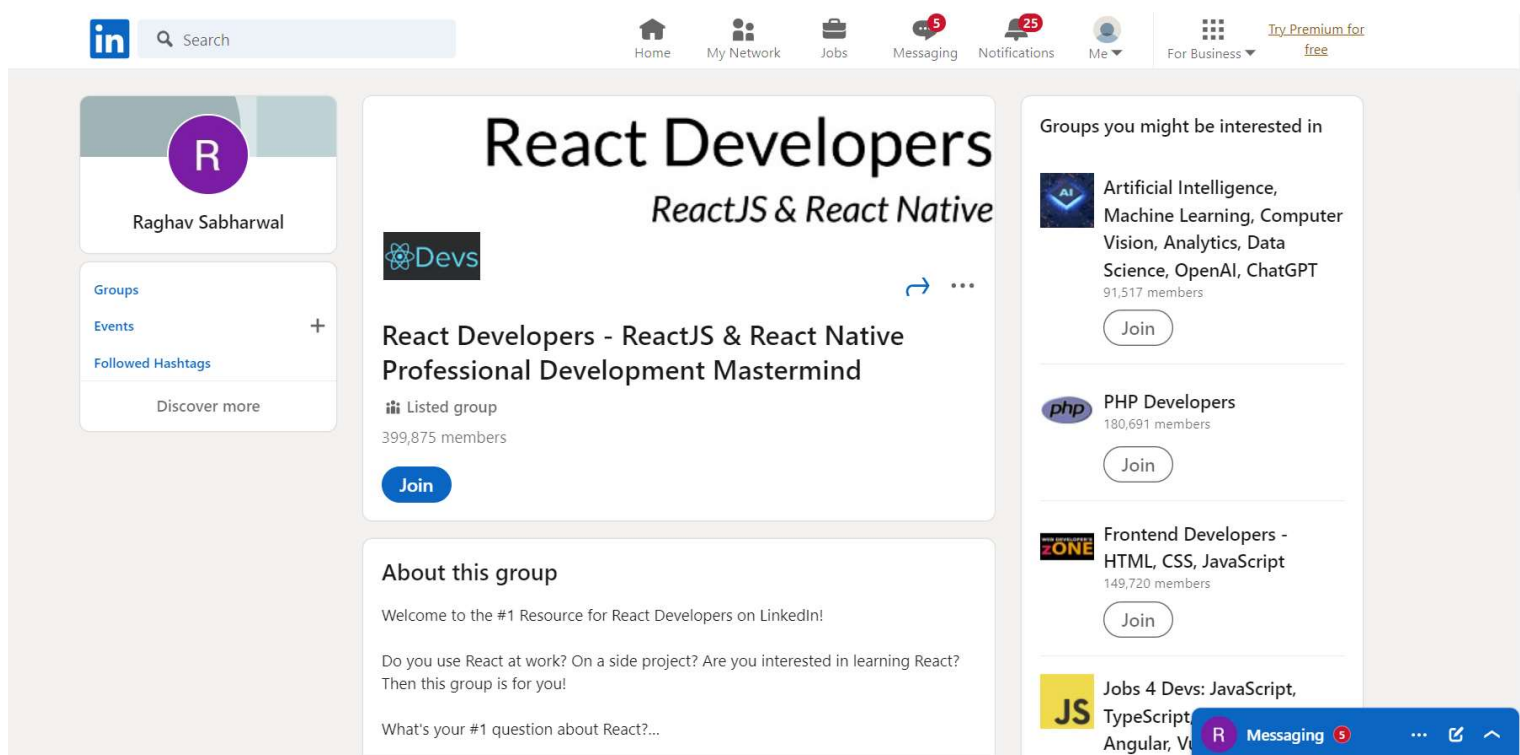
0/100

Description\*

What is the purpose of your group?

Create

Create Group View



Group View

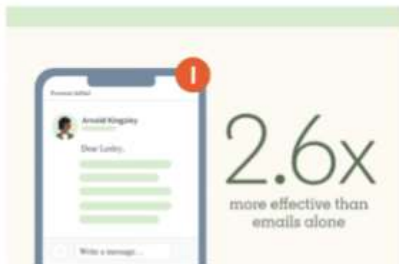
## Step 09: Subscription Page

- 👉 Once the user clicks on the Try Premium link, it should redirect to this page.
- 👉 Provide users with premium features or benefits for purchasing LinkedIn Premium. The UI should look like this.

**Get hired 2x faster on average** by discovering over 20 million open jobs and exploring valuable resources to help with your search.

[Try now](#)

#### InMail credits



#### Connect with hiring managers

Show your interest in an open role with InMail. It's 2.6x more effective than emails alone.

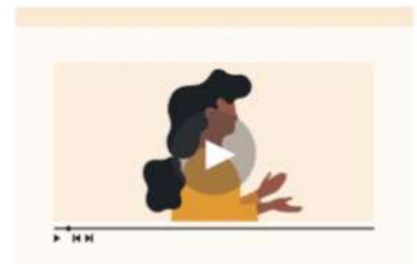
#### Who's Viewed Your Profile



#### Turn views into opportunities

See who's viewed your profile over the last 90 days, and who looks next.

#### LinkedIn Learning courses



#### Sharpen your skills

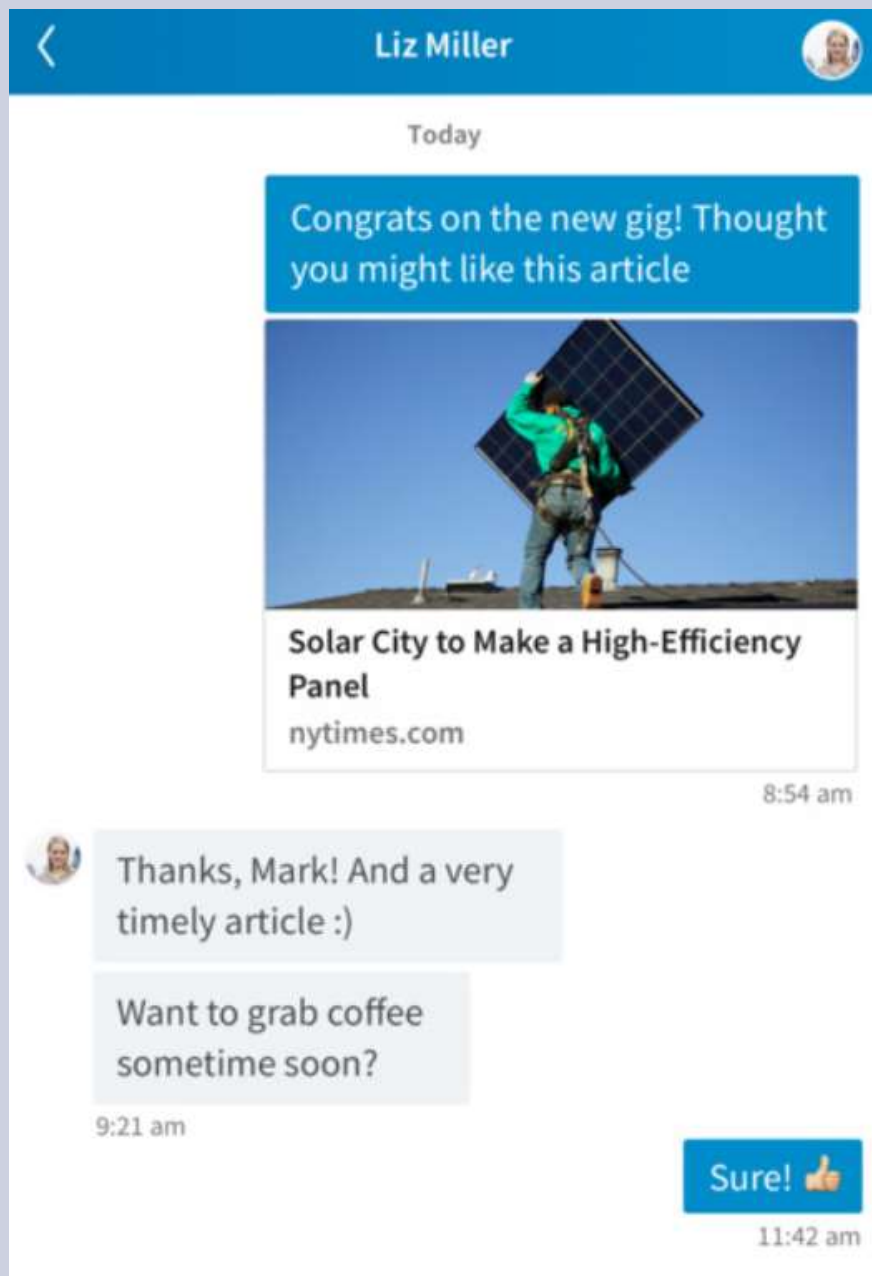
Hone your skills or try something new — access over 15,000 expert-led LinkedIn Learning courses.

© 2020 LinkedIn

## Step 09: Messaging and Chat

👉 Create a messaging system that allows users to send and receive messages from their connections.

👉 Enable chat functionality for a seamless messaging experience.



## Project Checkpoints

- 👉 Set up the project and UI design, Nav bar, Homepage
- 👉 Post Component, Search Functionality
- 👉 Login/ Signup page, Compose a Post

👉 User Profile and Networking/ View Group, Subscription Page, Messaging and Chat

ON THIS PAGE

---

Project Steps

