PROJECT MANAGEMENT TOOL

Collaborating tool for people

to work virtually

Installation Guide

TEAM 1: AVENGERS ASSEMBLE

About: What is Project Management Tool

Project management tool is an aid to assist an individual or team to effectively organize work and manage projects and tasks with various features for teams of various sizes. We aimed to develop the Project Management Tool, which helps an individual, team, or organization effectively organize work, projects, and tasks. This tool will have advanced features to track projects, Save and view their work, communicate with each other and much more. This is a working prototype of a management tool consisting of various features for collaboration, sharing and management of various tasks.

INSTALLATION GUIDE:

Installation guide for the project management tool will include some environment settings both on the front end and the back end. To set up the frontend we need to install React and NodeJs. For backend we need to install Python along with Django framework. Please follow the following steps to set up the environment on a local system. You can get the full code repository from the GitHub by following the link: Project Management Tool

In order to run the tool successfully we need to download all the dependencies to be downloaded in the local system:

- React
- NodeJS
- Code editor (preferably Visual Code Studio)
- Python (version 2.7 +)
- Django & Djangorestframework

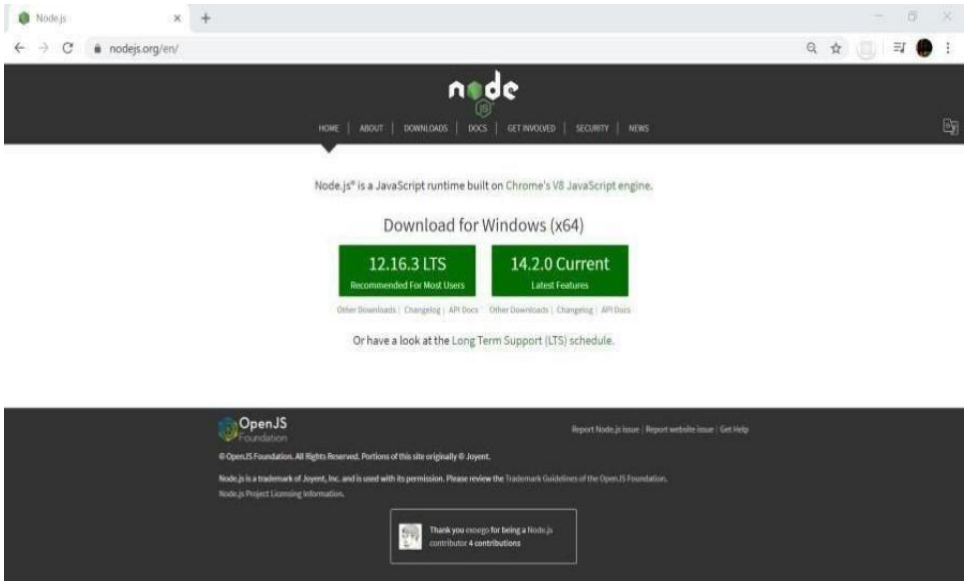
SETTING UP THE ENVIRONMENT*:*

- **Install NodeJS and NPM (Windows)**

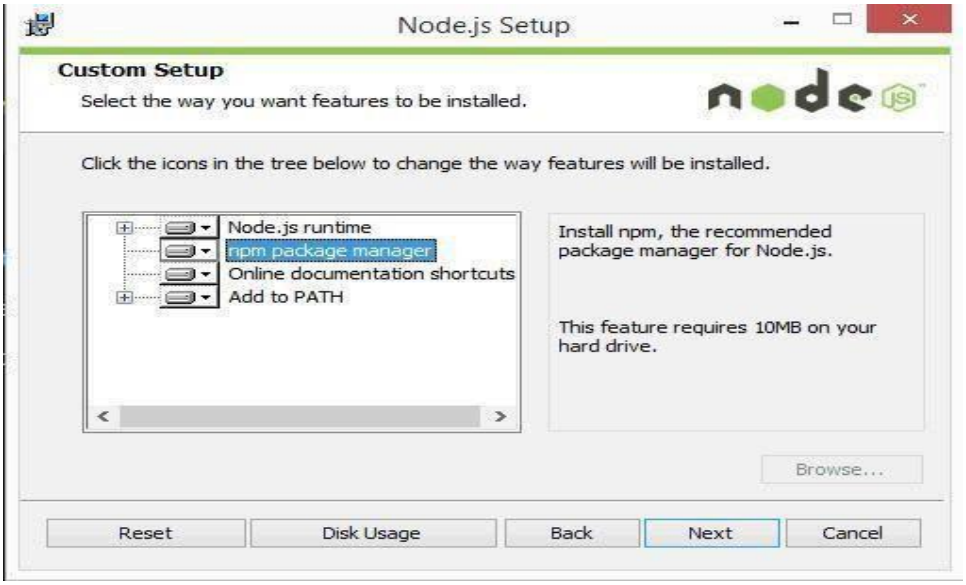
  Installing Node and NPM is using the installer package available from the Node.js® website.

  - **Installation Steps**

    1. Download the installer from the **Nodes.js® web site** for your operating system.



2. Run the installer (the file you downloaded in the previous step.)
3. Follow the prompts in the installer (Accept the license agreement, click the NEXT button and accept the default installation).



4. Restart the system (Recommended)

○ For Mac/Ubuntu

- Enable the NodeSource repository by running the following `curl` command as a user with sudo privileges :

```
$ curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
```

- Once the NodeSource repository is enabled, install Node.js and npm by typing:

```
$ sudo apt install nodejs
```

○ Test NodeJS and NPM

Make sure you have Node and NPM installed by running simple commands to see what version of each is installed and to run a simple test program:

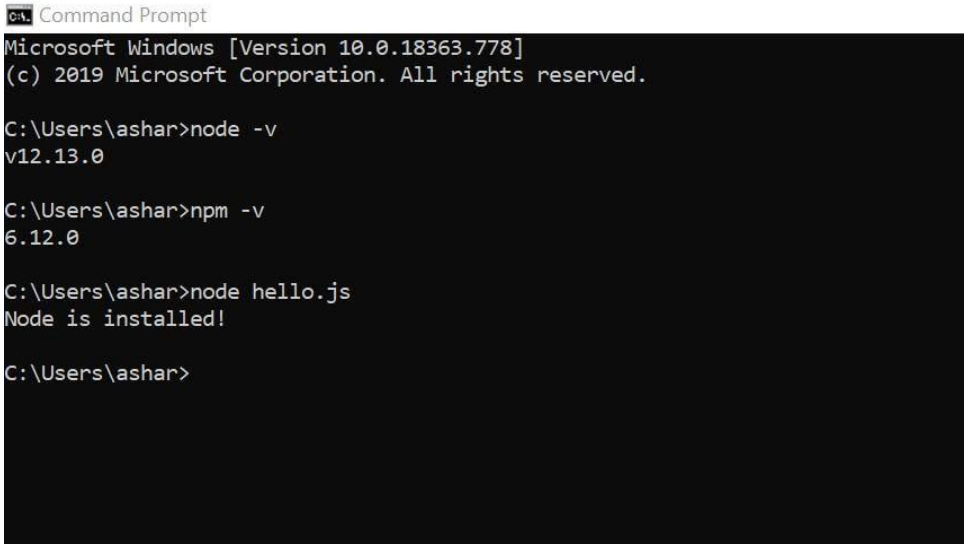
- **Test Node.** To see if Node is installed, open the Windows Command Prompt, PowerShell or a similar command line tool, and type ***node -v***. This should print a version number, so you'll see something like this v14.2.0 if you installed the latest version. In my case it is v12.13.0. Recommended version is v12 or above
- **For Mac/Ubuntu**

```
$ node --version
```

- **Test NPM.** To see if NPM is installed, type **npm -v** in Terminal. This should printNPM's version number, so you'll see something like this 6.5.2
- **For Mac/ubuntu**

```
$ npm --version
```

- **Create a test file and run it.** A simple way to test that node.js works is to create a JavaScript file: name it hello.js, and just add the code console.log ('Node is installed!'). To run the code simply open your command line program, navigate to the folder where you save the file and type node hello.js. This will start Node and run the code in the hello.js file. You should see the output Node is installed!



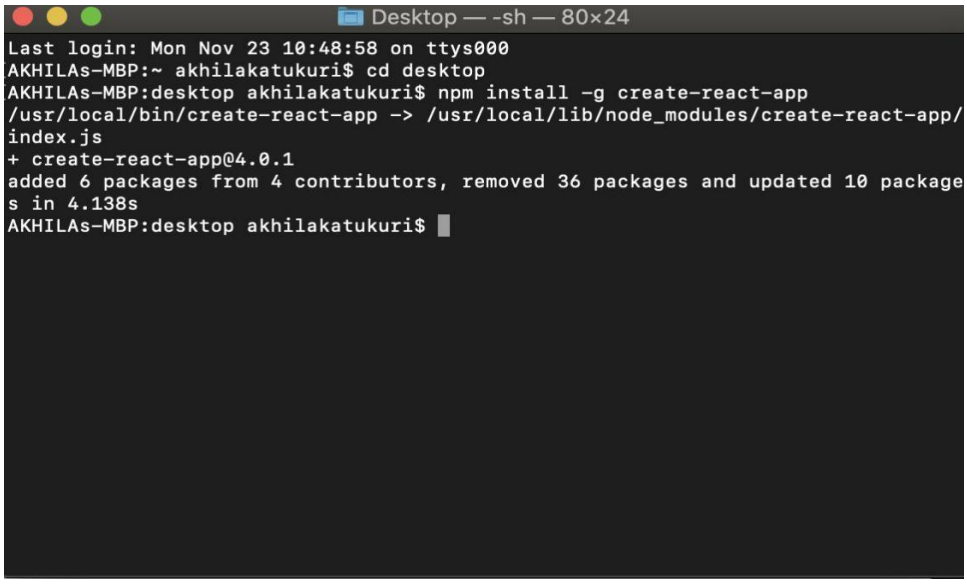
○ Install React and react DOM

- Our main task is to install ReactJS, install it, and its dom packages, using **install react** and **react-dom**commands of npm respectively. You can add the packages we install, to **package.json**file using the **--save**option.

- Setting up React Boilerplate. We will install the boilerplate globally. Run the below command in your terminal or command prompt to install the React Boilerplate.
  `npm install -g create-react-app`

- For Mac/Ubuntu
  `$ sudo npm -g install create-react-app`

1. After running the above command and successfully installing the boilerplate your terminal will show some output as shown in below image:



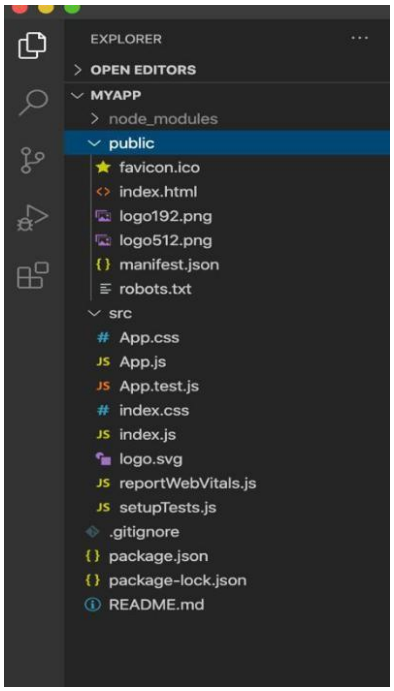
2. Create a react app. Now to create an app we will use the boilerplate we installed. Below command will create an app named *myapp*.

```
create-react-app myapp
```

For Mac/Ubuntu

```
$ create-react-app myapp
```

3. The above statement will create a new directory named *myapp*inside your current directory with a bunch of files needed to successfully run a React app.
Let's have a look directly created by the above command:



In the above directory, you can see a number of files. The main files we will be working with in the basic course are *index.html*and *index.js*.The index.html file will have a div element with id = "root", inside which everything will be rendered and all of our React code will be inside the index.js file.

Now, that we have successfully set up the development environment. The last thing left is to **start the development server**.

Start the development server. To start the development server, go inside your current directory "myapp" and execute the below command for both windows and mac/ubuntu :

```
npm start
```

You can go to the URL shown in the above message to see the changes you are making in your App. By default the above URL will show the below page:

○ Install Visual Studio Code

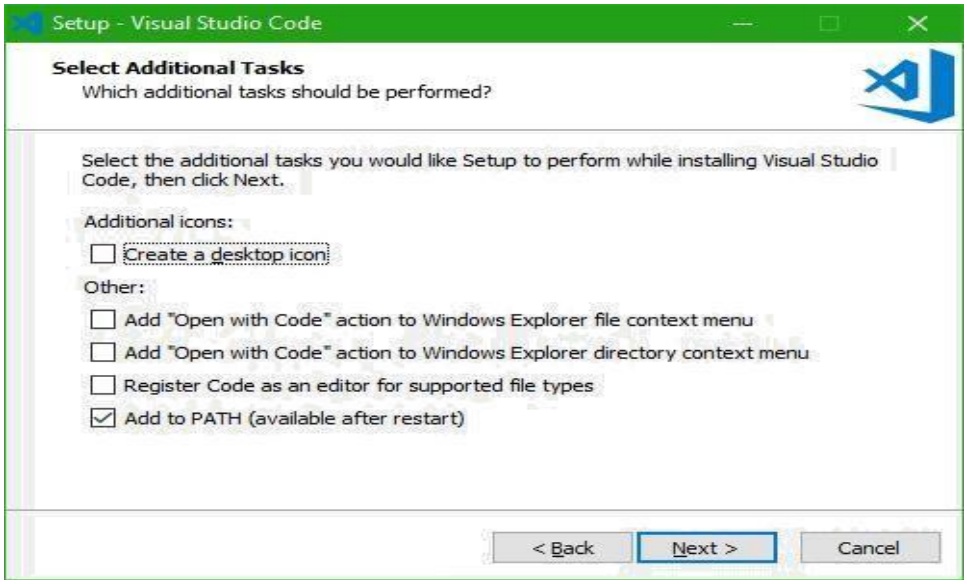
Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go).

Downloading Visual Studio Code is simple. You just need to head to this webpage and choose the package that matches your needs and requirements.

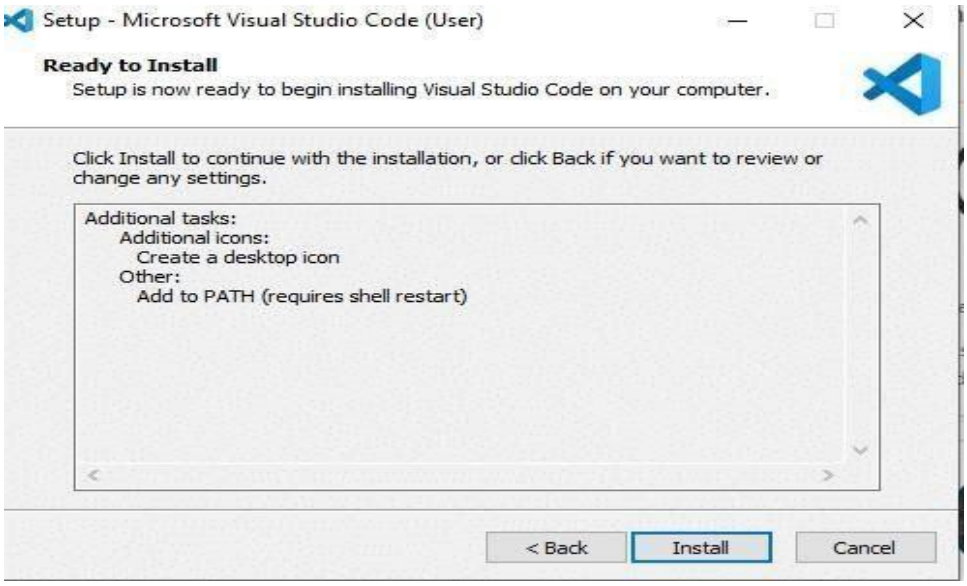
1. Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). This will only take a minute.
2. For Windows system, Click on Next.

3. Click on the I accept the agreement radio button and then hit Next and Next again and again.

4. In the following window, check “add to PATH” (available after restart)

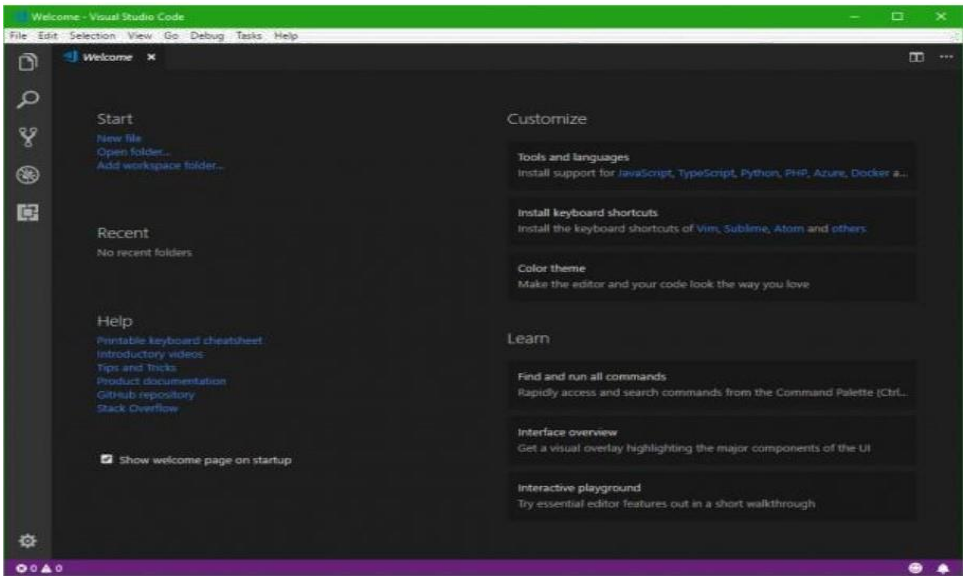


5. After that, click on the install button



6. Finally, after installation completes, click on the finish button, and the visual studio code will get open.

7. A new window will open for the Visual Code Studio



○ Installing Docker

Docker Desktop is an easy-to-install application for your Mac or Windows environment that enables you to build and share containerized applications and microservices. Docker Desktop includes Docker Engine, Docker CLI client, Docker Compose, Notary, Kubernetes, and Credential Helper.

Docker Desktop is available for Mac and Windows. For download information, system requirements, and installation instructions, see:

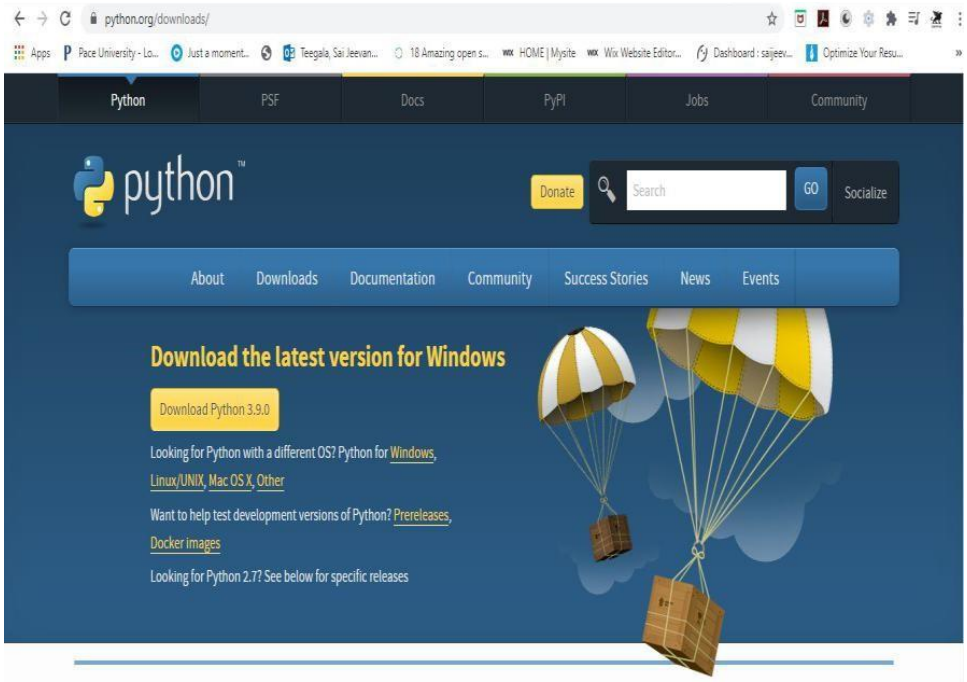
- Install Docker Desktop on Mac
- Install Docker Desktop on Windows

○ Install Python and Django

Django is a framework of python.
Installing Python is using the installer package available from the official python web site.

➢ Installation Steps

1. Download the installer from the https://www.python.org/downloads.You can select the operating system according to your choice with the latest version. We are currently using the python 3.7 for this project.



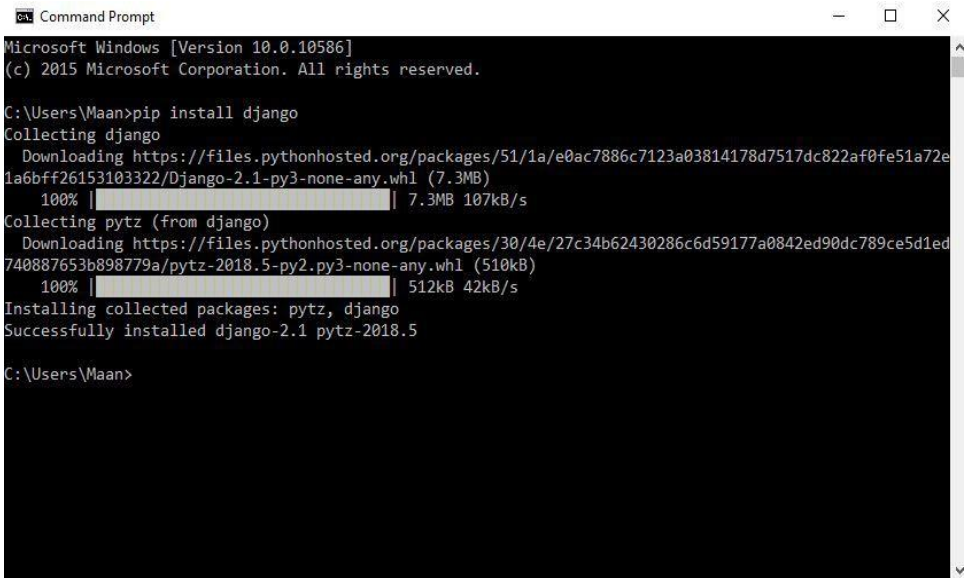
2. Run the installer (the file you downloaded in the previous step.)

3. Follow the prompts in the installer (Accept the license agreement, click the NEXT button and accept the default installation).

4. Restart your computer. (Recommended)

○ Install Pip and Django

- PIP is a package manager for Python that uses the Python Package Index to install Python packages. PIP will later be used to install Django from PyPI.
- Open a command prompt and execute **install pip**. This will install **pip** on your system. This command will work if you have successfully installed Setuptools.
- For Mac/Ubuntu run this command:
  ```
  $ sudo apt install python3-pip
  ```
- Django can be installed easily using **pip**.
- In the command prompt, execute the following command: **pip install django**. This will download and install Django
- For Mac/Ubuntu run this command:
  ```
  $ sudo apt install python3-django
  ```
- After the installation has completed, you can verify your Django installation by executing **django-admin --version**in the command prompt/terminal.



○ Setting Virtual Environment

➢ After installing django we need to set up the virtual environment.The development environment is an installation of Django on your local computer that you can use for developing and testing Django apps prior to deploying them to a production environment.

- You can install the virtualenv package using this command
  `pip install virtualenv`

- Then go to your project folder using `cd folder_ name` and try to run the Command
  `virtualenv virtual_environment_name`

- To activate the virtualenv you have to be in the project folder and then run this command to activate the virtual environment
  `virtualenv venv\Scripts\activate`

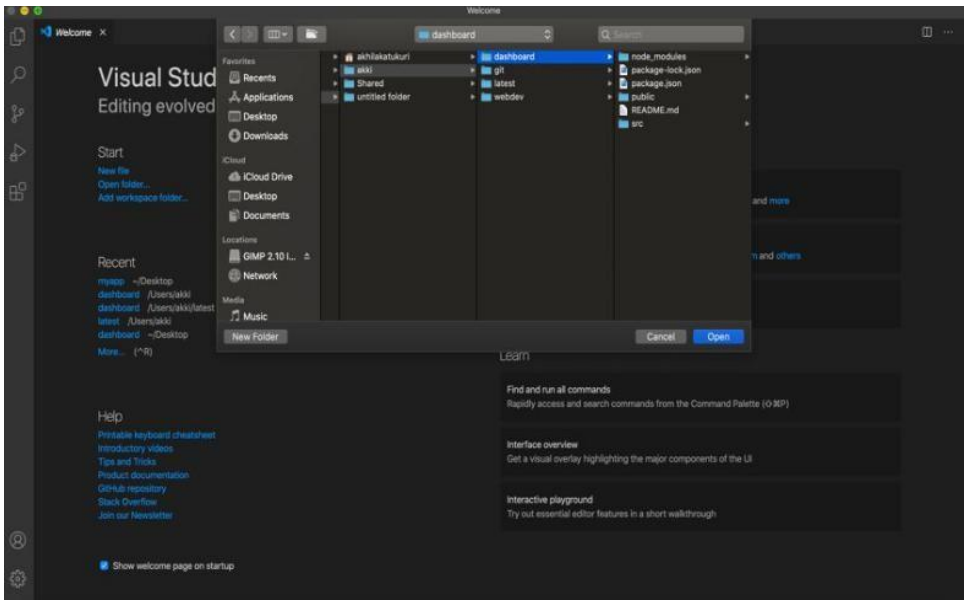
- Installing virtualenvwrapper-win is even simpler than setting up *virtualenvwrapper* because you don't need to configure where the tool stores virtual environment information (there is a default value). All you need to do is run the following command in the command prompt:
  `pip3 install virtualenvwrapper-win`

○ Starting the Application

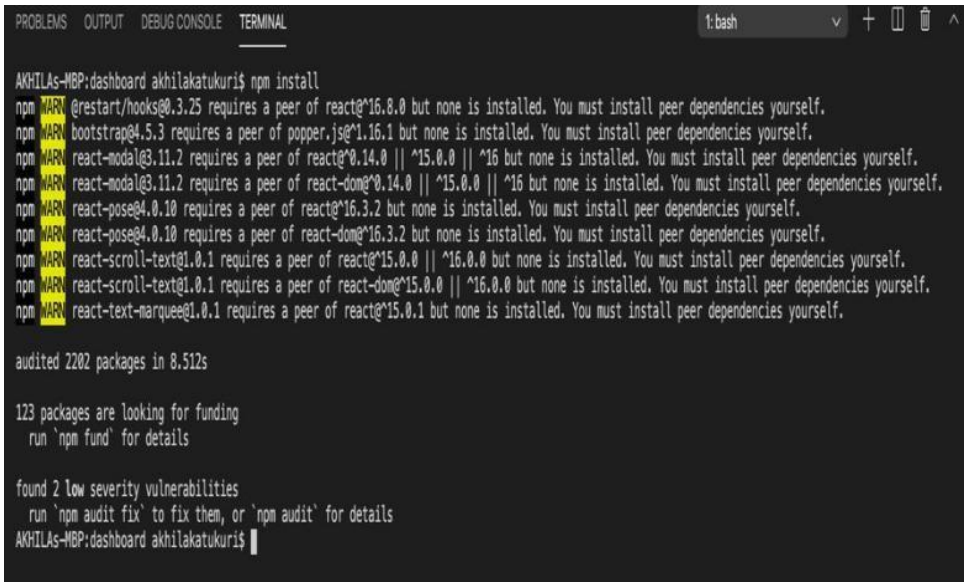
• Run User Interface

1. Download Project code from: Project Managment Tool and save in a folder. OR you can also clone the project if you have git bash using the following command:
   `$ git clone https://github.com/Vamshi399/project_management_tool`

2. Open WebStorm or VS code and open the project folder where you downloaded or cloned. It will look like this:



3. After loading the project, go to the “server” folder and run *npm install* in VS code Terminal.



4. Run the npm dev to start the Application in the VS code terminal. After running successfully, the server will start on the https://localhost:3000’

5. Now open a second terminal to initialize docker for chat server and run the following command to check the dependent requirements:
```
pip install -r requirements
```

6. Now after getting all the requirements turn on the docker from the application and then run the command in the VS code terminal
```
docker run -p 6379:6379 -d redis:5
```

7. Now to start the backend server we need to open a second bash terminal. After this go the the client folder by using the command:
```
$ cd/client
$ cd/ wframe
```
8. Now when in the wframe, activate the virtual environment by the following command:
```
virtualenv venv\source\activate
```

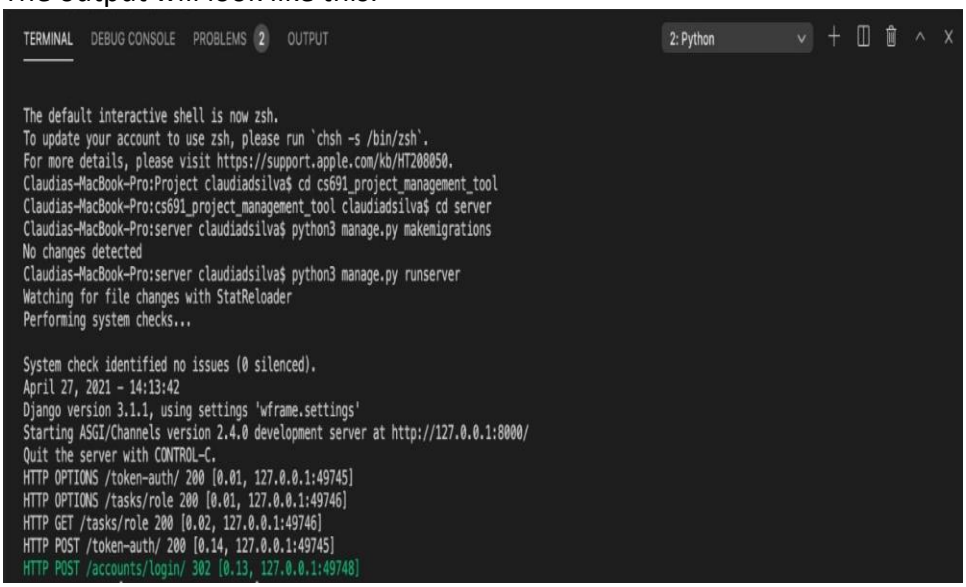
   for mac/ubuntu
```
virtualenv venv\bin\activate
```

9. Then run the following commands to see if any migrations are needed
```
$ python manage.py makemigrations
$ python manage.py migrate
$ python manage.py check
```

10. After making the necessary migrations run the server by using the command:

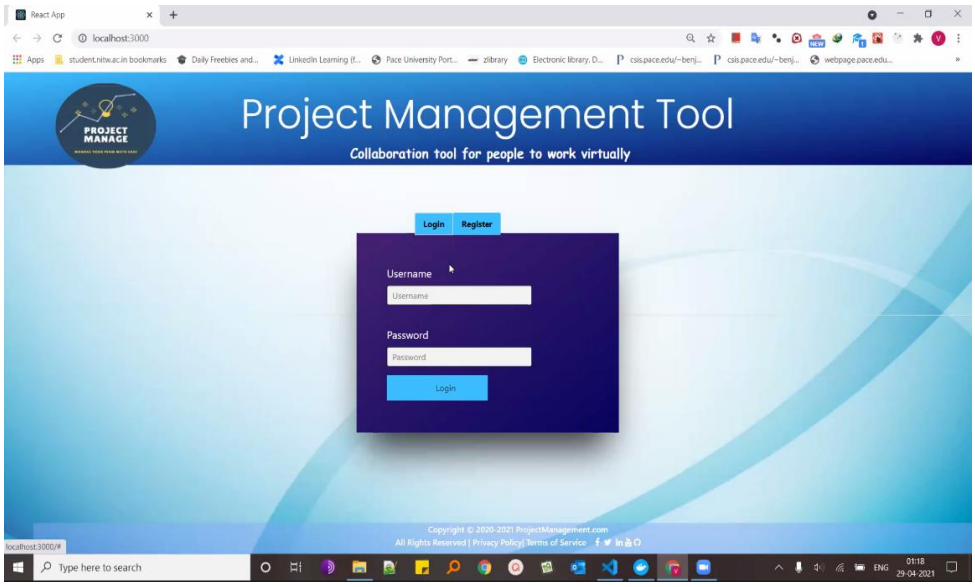
```
Python manage.py runserver
```

The output will look like this:

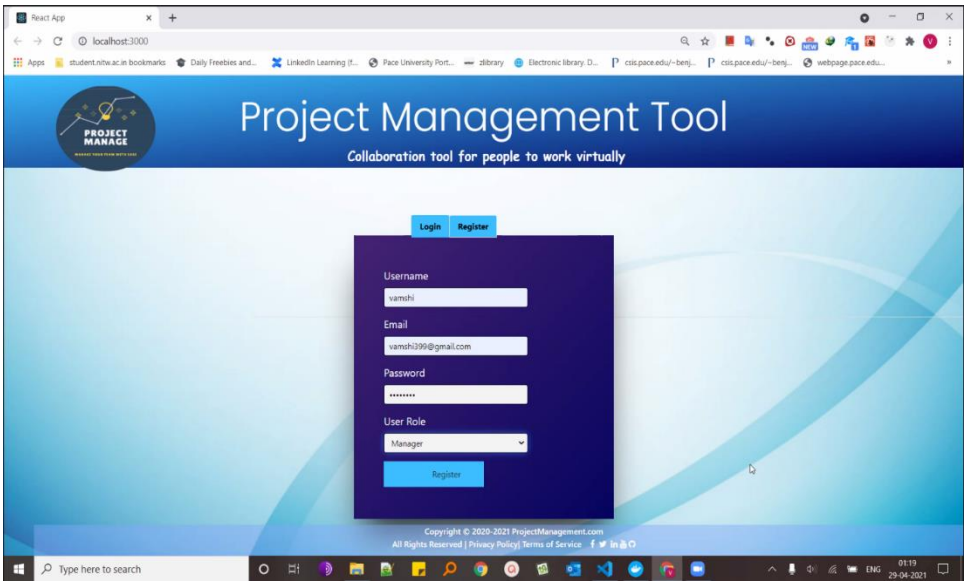


○ Using the Application

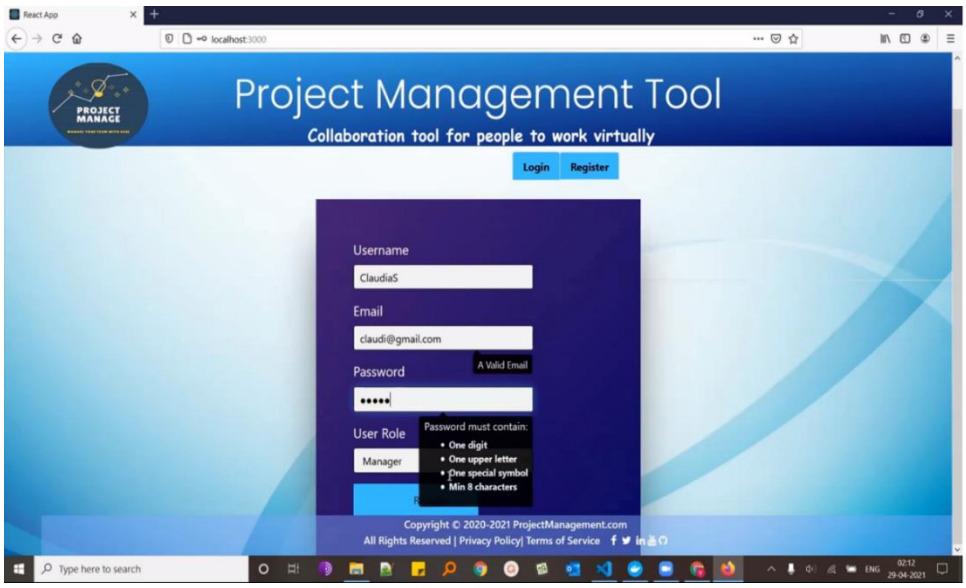
- The Login page will look like this



- Now you can Register and login to the application

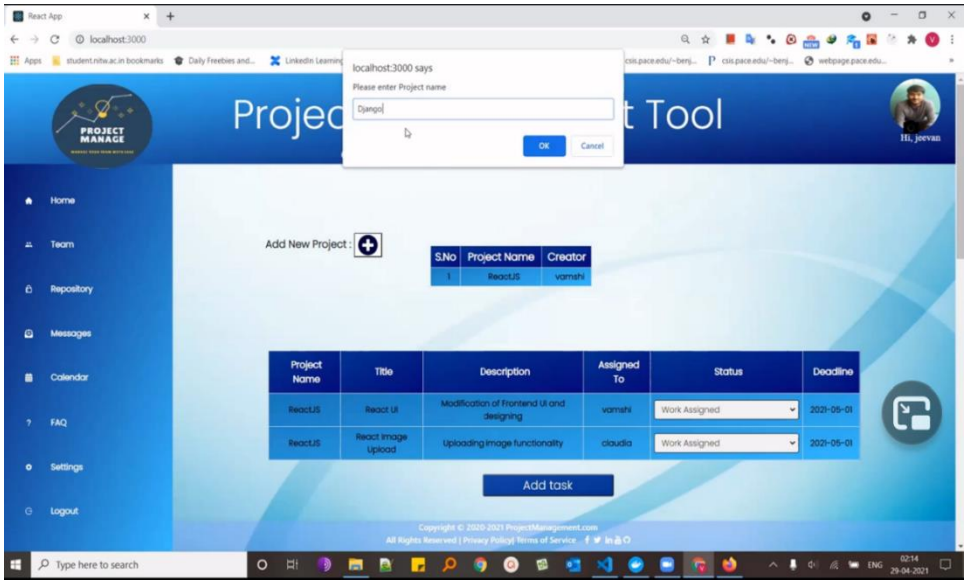


- Enter valid Email and password to register

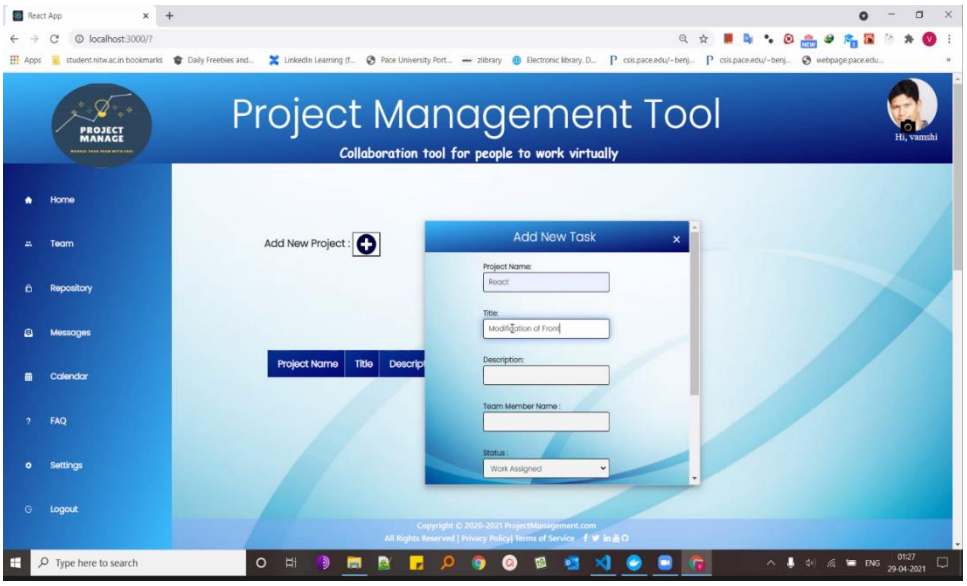


- Once you login you'll see the dashboard

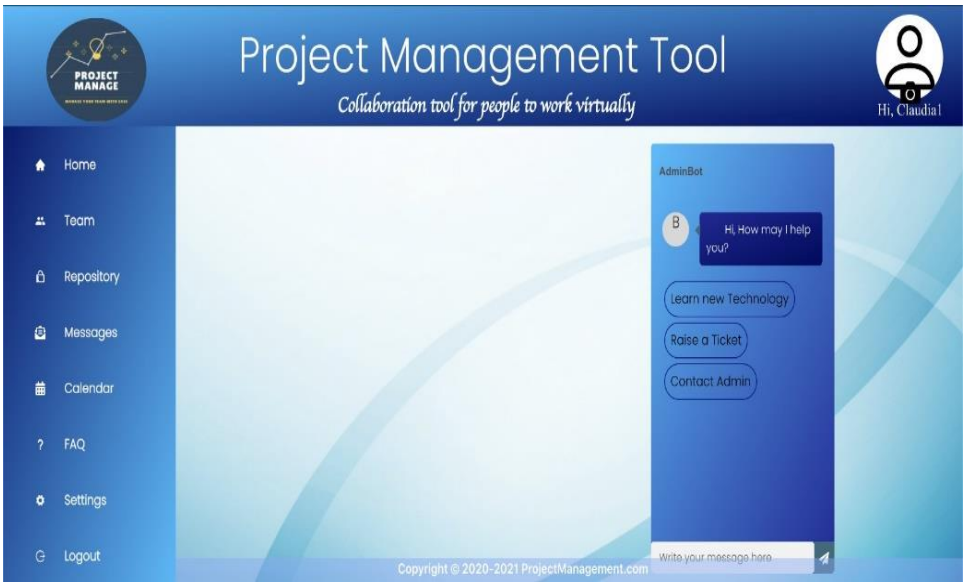
- When you click on add new project, a popup is displayed with Enter Project name. If you have added a project it'll look like this and no project is added there will be nothing on the screen.
  Enter project name and click ok. Now the added project will be displayed in project table.



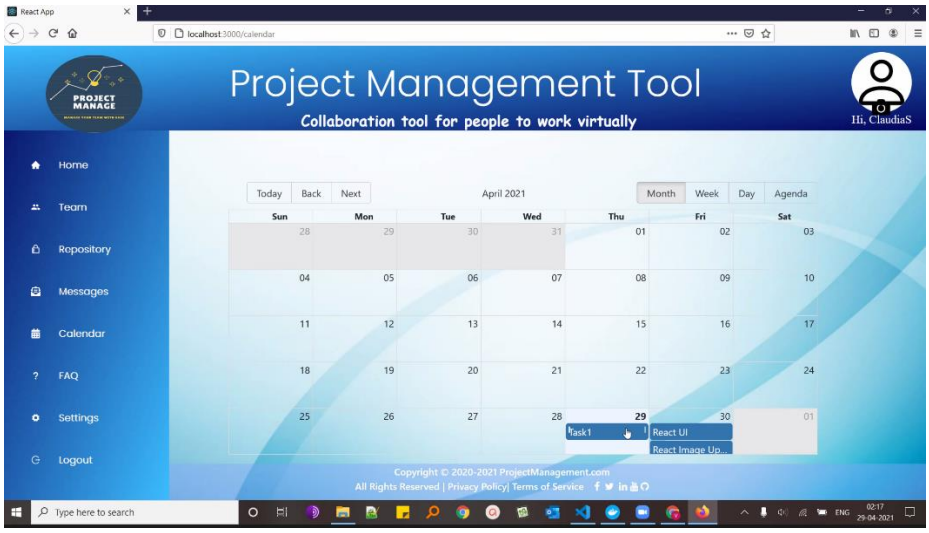
- Click on add task button to open the modal. Enter all the details and click on add item to add a new task.



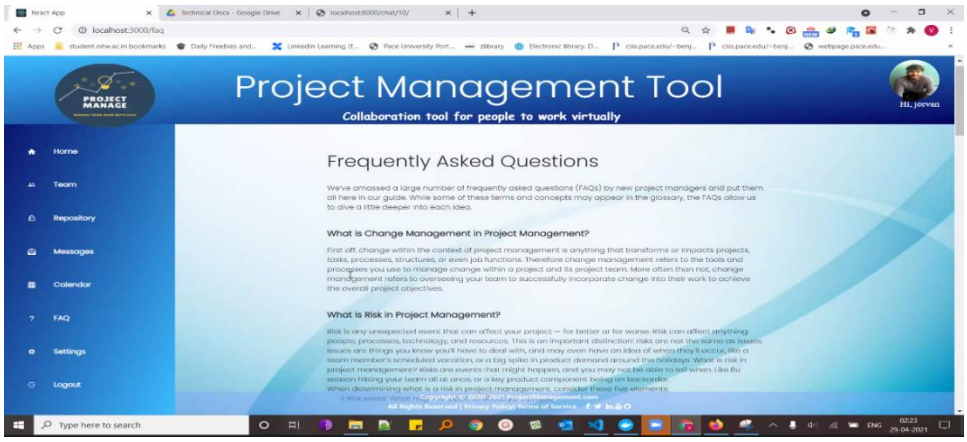
- On the Home tab you can also see the chatbot at the bottom of the screen



- Click on Calendar to view the calendar and add notes.



- FAQ page will also give you some more information



- Some more settings to personalize the account can be seen in the Settings page