MAJOR PROJECT REPORT ON

ProjektHouse – A Web platform for students for sharing projectsand ideas.

Submitted for fulfillment of award of

Bachelor of Technology

In

Information Technology

Submitted by

Saksham Gupta

Sanyam Gupta

Uddeshya Sonkar

Sabyasanchi Bharadwaj

Under the guidance of

Mr. Sumit Sharma Sir

Ajay Kumar Garg Engineering College,Ghaziabad
Year 2021-2022

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ajay Kumar Garg Engineering College,Ghaziabad

Year 2021-2022

Certificate

Certified that we have carried out the Project work entitled “ProjektHouse” for the award of Bachelor of Technology from Ajay Kumar Garg Engineering College, Ghaziabad under our supervision.
To the best of our knowledge, this work has not been submitted earlier to any university for the award of any degree

Guide Name Dr. Anu Chaudhary
(Designation) Head of Department
Project Guide Information Technology

Acknowledgement

would like to express our sincerest gratitude to all the people who have contributed towards the successful completion of our project.

We would like to extend our heartfelt thanks to the Head of Information Technology Department Dr. Anu Chaudhary, for nurturing a congenial yet competitive environment in the department, which motivates all the students to pursue higher goals.

We want to express our special gratitude to our guide “Guide name, Guide Designation”, Department of Information Technology, Ajay Kumar Garg Engineering College, Ghaziabad for his/her constant support, guidance, encouragement and much needed motivation. His/Her sincerity, thoroughness and perseverance has been a constant source of inspiration for us.

Last but not the least, we would like to extend our thanks to all the teaching and non teaching staff members of our department, and to all our colleagues who helped us in completion of the project.

1. Saksham Gupta

2. Sanyam Gupta

3. Sabyasanchi Bharadwaj

4. Uddeshya Sonkar

Abstract

Social media today plays an expanding significant role in society, the information technology industry and the field of computer science. The use of social media is a hot-topic for many organizations, with the aim to identify approaches in which companies can use applications to increase profits and grow product awareness. On a day-to-day basis, users from across the globe are becoming increasingly frustrated, wasting valuable time, scrolling through irrelevant content while companies are wasting money advertising to users outside their market. In order to achieve the optimal benefits from social media, for both users and businesses, the development of these technologies require approaches that focus on specific human interests and values.

There are platforms where professionals can share their latest research works or career updates and discuss such things with others. But there has always been something missing exclusively for students to share their projects and ideas and research paper ideas and get feedback from students all around . This project aims to deliver the solution by developing a platform with a goal of providing features such that students could easily discuss ideas and projects and find like minded people on the platform to work on .

With the whole buzz of social networking around in the world , students also need a platform for sharing their projects and transforming some of the ideas into something great .

MOTIVATION

We in our student life make a lot of projects that just stay in our local devices and never reach to others for reviews and real time usage .A lot of students make projects , so to make them able to share it with all others and get reviews and learn new things ,We thought of this platform. With this Professors will also linkmore with students and students at the same time can collaborate and learn more.

With the platform launching everywhere , this will encourage a lot of students to work together and make some great projects, not just leaving them in their local devices . The idea of sharing and building a large scale project motivated me to build something like this .

.

Table of Contents

Chapter-1

Introduction

1.1. Problem Overview

There are multiple platforms for connecting and sharing our day to day life updates . Even there are platforms like LinkedIn and Research gate , for sharing career updates and research papers respectively . But there is not any dedicated platform for students to share their project works and ideas that they make during their course of college .
We as students make a lot of projects and they just stay in our local devices most of the time and come of no use and we don't even get feedback on shortcomings and what features should be added etc . This may seem like nothing , but a huge issue . If there would be a platform for sharing projects and ideas with all around , students can get feedbacks and better reviews of their projects and they can get huge reach , also most of the startups begin as a project in a group or as an individual group , a lot of projects are made but only a handful reaches to the top . To solve this problem and help students reach volumes and turn their projects and ideas to something big , a platform dedicated for the same is much needed .

1.2 Objectives

To solve the problem statement , I have come up with this solution , **ProjektHouse .** A web platform dedicated for the students to showcase their projects and ideas and research papers and get feedback from students all around and get better reviews and reach masses. A web based app like LinkedIn or research gate but dedicated for students and focussing on the problem and helping students . The web app is built using the latest technologies with a lot iof features and a good looking interface . The idea helps students a lot and will encourage students to build better and more and come up with some great start ups of their own .
The web app will be based on MERN stack technology with hasslefree features and students dedicated aims .

1.3 Features

Application features -

- An easy to use single page wep application with responsive interface
- Full CRUD operations with self made APIs using express and node
- Fully functional and secure MongoDB database with restrictions and validations.
- Secured sensitive data with encrypting properly .
- Servers hosted on heroku
- Built with the best and latest tech stack.

User features -

- Can register an account hassle free in a single step
- Can login any time from any device without any issue
- Can stay logged in using the local storage
- Can log out of the website anytime
- Can update the profile image to whatever wanted
- Can follow/unfollow the creators on the platform
- Like/ Comment on post
- Exclusive video call and chat feature at 3rd party web application
- Can add post with or without photos

1.4 Methodology

The project is based on the latest web technology , i.e. MERN stack (MongoDb, Express js, React js , Node js) . Project will be held on 2 different servers , one being the front end and other being the backend server . Both servers will run simultaneously and work hand in hand .
The back end server is based on Node js and the front end server is based on React js . The Database used is MongoDb , which is non-relational , secure and better to use .
All the frameworks , libraries used are Open -Source and free to be optimised and used .
The editor used to code is VS Code and for management of files and updates , github is used extensively .
At the end , Heroku is used to host the project online and provide everyone hasslefree service .

All the code is done from scratch and no code is copied from any online resources, for better handling of course , Material framework is used in the Front End part .

Chapter-2

LITERATURE SURVEY

LinkedIn is a similar platform , for the professional people and students too , to share their career updates mainly . This platform has a huge user base and almost all of the students and professionals in all the fields are there with the aim of looking out for new opportunities and meeting new professionals and companies there .

It is a very useful platform for all the students too , as they get to know about the latest openings of jobs in various fields and get a chance to apply for the jobs and chance of securing a job .

In the past it also has been used as a platform for sharing ideas and projects and related work with others , but the platform is more of a professional platform and thus asks one to be the same and thus we don't meet more of this kind of students who are there for projects and ideas .

Hence it becomes important to build such a platform which has similar features to it , but dedicated to students and building an environment around , so that students can meet other students with similar minds and can work on something similar .

Some of the important features are -

- Posting about jobs and career opportunities in various companies
- Sharing projects and achievements
- Built using Java
- Misses a separate dedicated section for students as described above in the problem statement .

Similarly , there is ResearchGate .

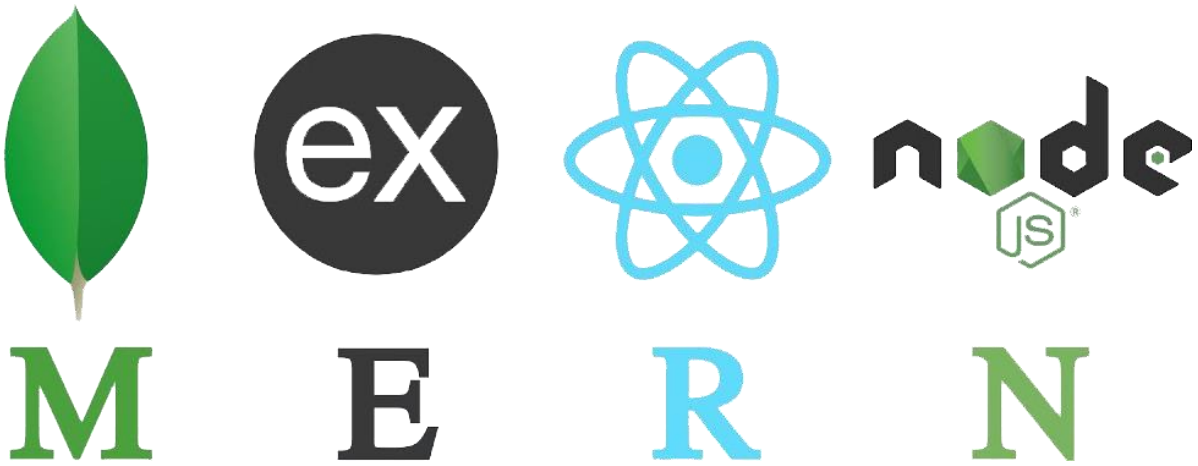
ResearchGate is more of a professional platform for research students and professors to share their research papers . It is mainly dedicated to professionals and sharing their cited papers and research works .

But ResearchGate also is more of a professional platform rather than a student dedicated platform for sharing their ideas and projects .

Both the platforms have millions of users around the globe , and a similar platform specially dedicated to students development can also get such a huge user base .

Chapter-3

SYSTEM REQUIREMENT & TECHNOLOGY STACK



To develop this web application , I used mainly th MERN web stack , i.e. MongoDb + Express js + React Js + Node js .

A brief list of some technologies used : -

- **Languages**
  - JavaScript
  - HTML
  - CSS
- **Libraries**
  - Mongoose
  - Bootstrap
  - Reactjs Material
- **Frameworks**
  - Express
  - React js

- **Database**
  - MongoDB
- **Environment**
  - Nodejs
- **Development Software**
  - Visual Studio Code
  - Postman
  - Git

Below I have explained every part in detail -

ReactJs

React is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.
React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality .
It makes the web application faster and smoother , it provides hot reloading and multiple other features to make the platform better and faster .
It is the most used technology for the front end part these days .

Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine . It is an open-source, cross-platform, back-end Javascript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser . Node.js lets developers use JavaScript to write command line tools and for server side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,unifying web

application development around a single programming language, rather than different languages for server-side and client-side scripts.
Node js is used on the backend side of the code for the backend server with Express js.

ExpressJs

Express is a minimal and flexible Node.*js* web application framework that provides a robust set of features for web and mobile applications. Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.
It has been called the de facto standard server framework for Node.js.

Express is the back-end component of popular development stacks like the MEAN, MERN , MEVN stack, together with the MongoDB database software and a Javascript front-end framework or library.

MongoDB

MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era. MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License.
It is non relational , which makes it better than SQL and is used in the popular development stack such as MERN , MEAN and MEVN .

Chapter 4

Project Codes -

Whole project and the files were managed using github and heroku and local devices .

Deploying the project locally -

Prerequisites :

- MongoDB installed
- Nodejs installed
- git to clone the project
- Access to internet and internet browsers

- Download MongoDB

The project uses MongoDB. So it is much needed that local device has mongoDB installed. For detailed instructions to download and install , got to
https://treehouse.github.io/installation-guides/windows/mongo-windows.html

- Install required libraries

After cloning the project some libraries and decencies will be required. This can be achieved by installing and updating the node package manager.

Navigate to the downloaded repository and enter :

- npm install
- npm update

You have to do **npm install** , 2 times , once in the main directory and again in the client directory for front end .

- Now Run the development server

To deploy locally navigate to the project directory in cmd.
Now there are 2 servers to be run , the backend one and front end one .
Run the following command to build the project and launch the server:

- npm start (in the main directory)
- cd client/
- npm start (for front end)

The server will now be running and connected to MongoDB. Navigate to **localhost:3000** to view the application.

In the above way you can simply run the app on your local device .

Deployment

The application is currently deployed on HEROKU Server , which is free for limited users and can be found at - https://projekthouse.herokuapp.com/

Some of the main codes are -

Server side -

```
const express = require("express");
const app = express();
const mongoose = require("mongoose");

const PORT = process.env.PORT || 5000;
const { MONGO_URI } = require("./config/keys");

mongoose
  .connect(MONGO_URI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
    useFindAndModify: true,
  })
  .then(() => {
    console.log("MongoDB Connected");
  })
  .catch((err) => {
    console.log(err);
  });

require("./models/user");
require("./models/post");

app.use(express.json());
app.use(require("./routes/auth"));
app.use(require("./routes/post"));
app.use(require("./routes/user"));
```

```
if (process.env.NODE_ENV == "production") {
  //for heroku deployment
  app.use(express.static("client/build"));
  const path = require("path");
  app.get("*", (req, res) => {
    res.sendFile(path.resolve(__dirname, "client", "build", "index.html"));
  });
}

app.listen(PORT, () => {
  console.log(`Server Running on ${PORT}`);
});
```

Authentication -

```
const express = require("express");
const mongoose = require("mongoose");
const bcrypt = require("bcryptjs");
const jwt = require("jsonwebtoken");

const User = mongoose.model("User");
const { JWT_SECRET } = require("../config/keys");
const requireLogin = require("../middlewares/requireLogin");

const router = express.Router();

router.post("/signup", (req, res) => {
  const { name, email, password, pic } = req.body;
  if (!name || !email || !password) {
    return res.status(422).json({ error: "Please Add all the fields !!" });
  }
  User.findOne({ email: email })
```

```
    .then((savedUser) => {
      if (savedUser) {
        res.status(422).json({ error: "User already Exists" });
      }
      bcrypt.hash(password, 12).then((hashedPassword) => {
        const user = new User({
          name,
          email,
          password: hashedPassword,
          pic,
        });
        user
          .save()
          .then((user) => {
            res.json({ message: "User Saved Successfully" });
          })
          .catch((err) => {
            console.log(err);
          });
      });
    })
    .catch((err) => {
      console.log(err);
    });
});

router.post("/signin", (req, res) => {
  const { email, password } = req.body;
  if (!email || !password) {
    res.status(422).json({ error: "Please Enter both Email and Password" });
  }
  User.findOne({ email }).then((savedUser) => {
```

```
    if (!savedUser) {
      res.status(422).json({ error: "User Doesn't Exist with this email ID" });
    } else {
      bcrypt
        .compare(password, savedUser.password)
        .then((didMatch) => {
          if (didMatch) {
            const token = jwt.sign({ _id: savedUser._id }, JWT_SECRET);
            const { _id, name, email, followers, following, pic } = savedUser;
            res.json({
              token,
              user: { _id, name, email, followers, following, pic },
            });
          } else {
            res.status(422).json({ error: "Invalid Password" });
          }
        })
        .catch((err) => {
          console.log(err);
        });
    }
  });
});

module.exports = router;
```

Front -End Side -

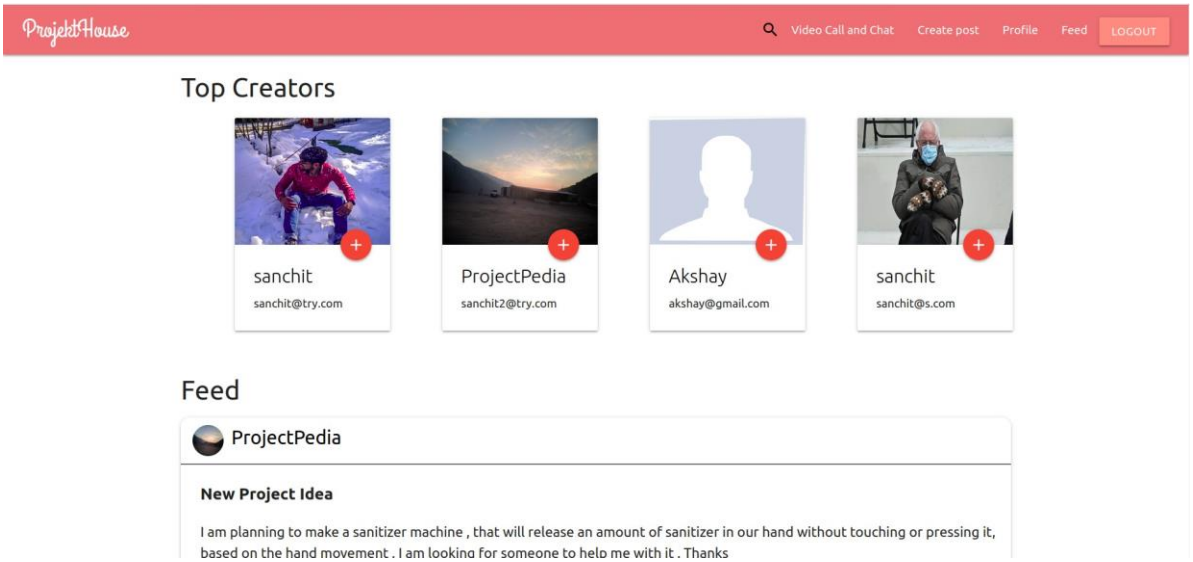
```
import React, { useEffect, useContext, createContext, useReducer } from "react";
import { BrowserRouter, Route, Switch, useHistory } from "react-router-dom";
import Navbar from "./components/Navbar";
import Home from "./components/screens/Home";
import Login from "./components/screens/Login";
import Profile from "./components/screens/Profile";
import Register from "./components/screens/Register";
import CreatePost from "./components/screens/CreatePost";
import { reducer, initialState } from "./reducers/userReducer";
import "./App.css";
import UserProfile from "./components/screens/UserProfile";
import SubsPosts from "./components/screens/SubsPosts";
export const UserContext = createContext();
const Routing = () => {
  const history = useHistory();
  const { state, dispatch } = useContext(UserContext);
  useEffect(() => {
    const user = JSON.parse(localStorage.getItem("user"));
    if (user) {
      dispatch({ type: "USER", payload: user });
    } else {
      history.push("/login");
    }
  }, []);
  return (
    <Switch>
      <Route exact path="/">
        <Home />
      </Route>
      <Route exact path="/profile">
```

```
        <Profile />
      </Route>
      <Route path="/profile/:userId">
        <UserProfile />
      </Route>
      <Route path="/login">
        <Login />
      </Route>
      <Route path="/register">
        <Register />
      </Route>
      <Route path="/createpost">
        <CreatePost />
      </Route>
      <Route path="/myFollowersPosts">
        <SubsPosts />
      </Route>
    </Switch>
  );
};
function App() {
  const [state, dispatch] = useReducer(reducer, initialState);
  return (
    <UserContext.Provider value={{ state, dispatch }}>
      <BrowserRouter>
        <Navbar />
        <Routing />
      </BrowserRouter>
    </UserContext.Provider>
  );
}
export default App;
```

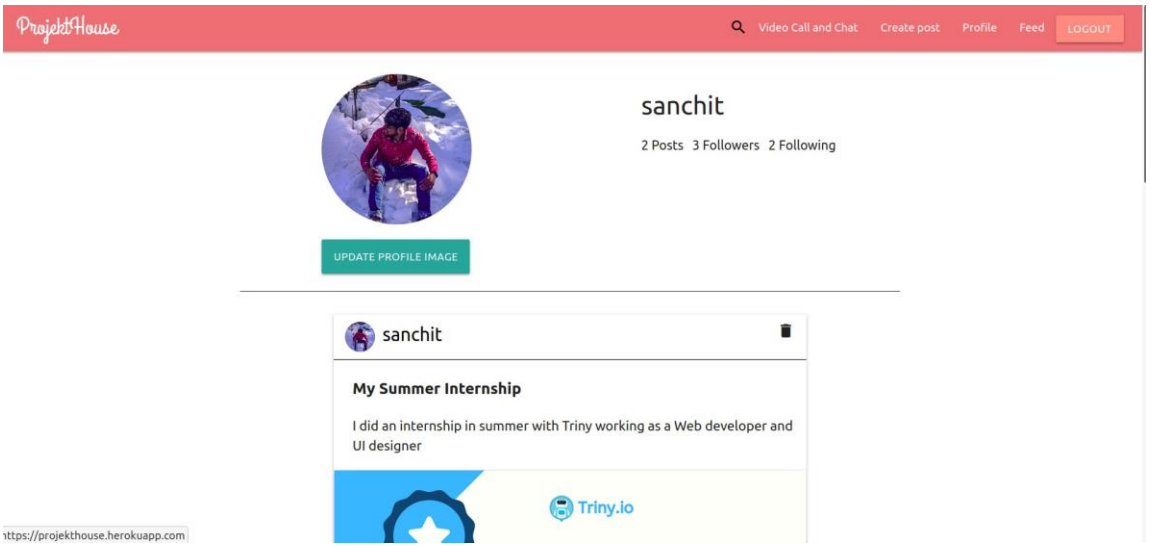
Chapter-5

Previews and Results

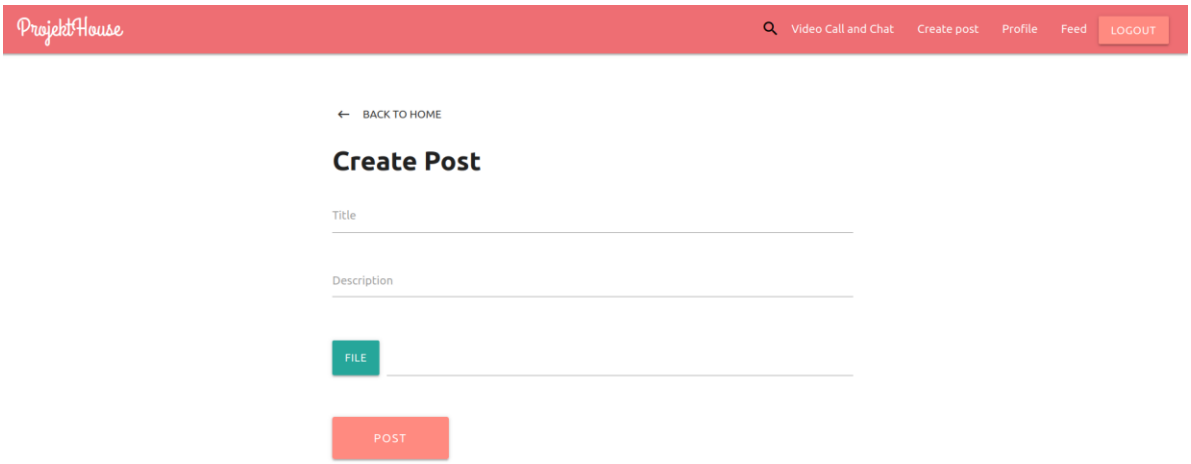
Home Page -



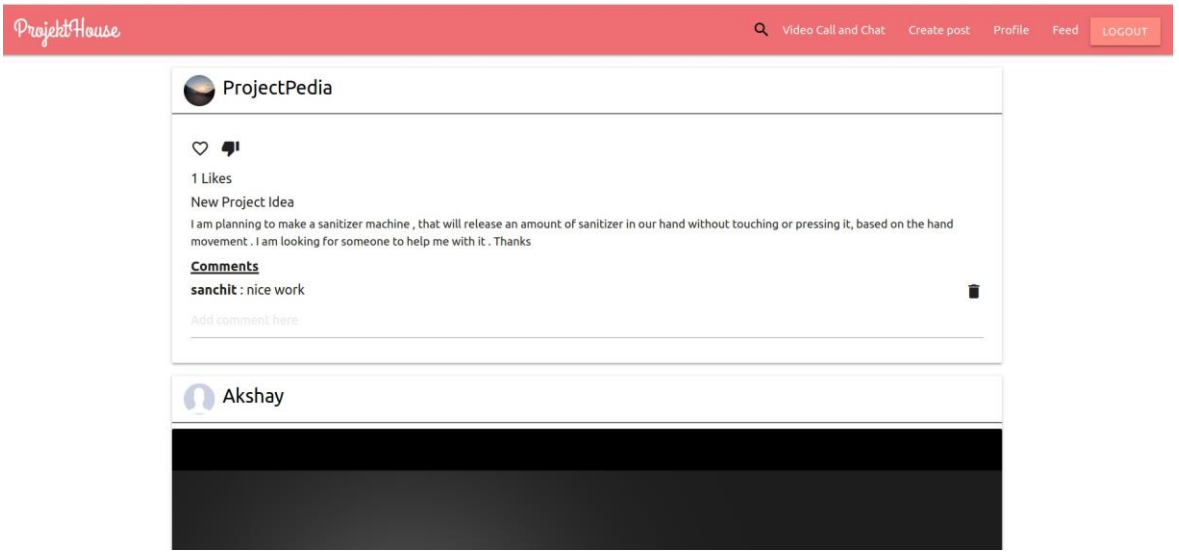
Profile Page -



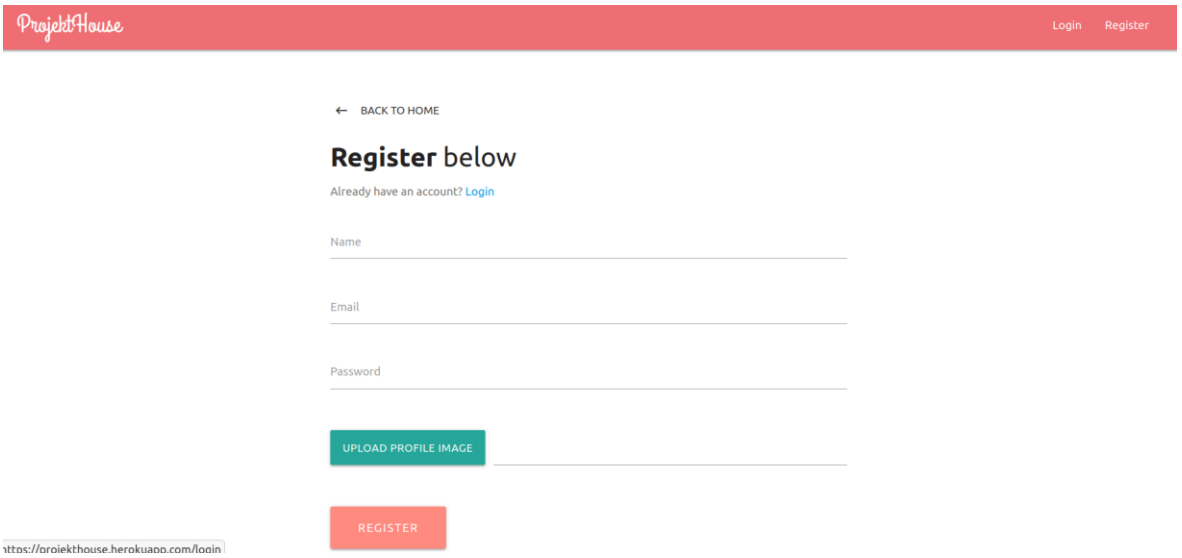
Create post Page



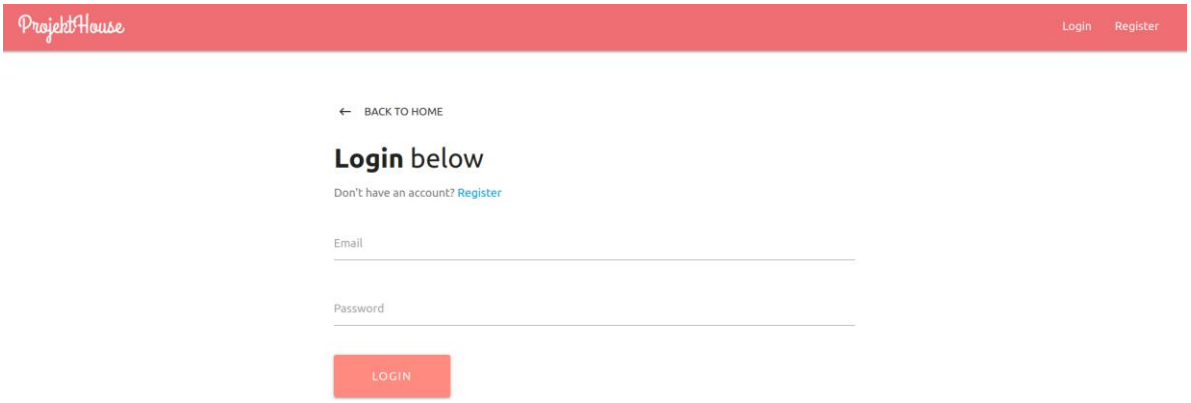
Following Post Page



Register Page



Login page



Above shown are the screenshots of the live app hosted on heroku https://projekthouse.herokuapp.com/ .

The web app works smoothly , currently only a handful of students are registered , but with scaling up , I aim to get a lot of students registered and make the platform live on a bigger scale with some more and better features and UI.

Future Works :

- Adding more features such as chat and video call within the application , currently it is not linked .
- Expanding it to a large scale and inviting other college students too.
- Hosting various events by known people in the tech industry.
- Improving security and layers of security .
- Hosting it on a paid service for better working . .

SUMMARY

With this project and platform ,We aim to help students and encourage them to build better and more . This can react to a lot of students and can have quite a big user base . In the future , it holds a lot of scope , It would be usedby a lot of students to get feedback and reviews . It will come out to be the best secured and dedicated platform for the students , by the students.

The platform will focus towards building and making ideas and projects of students to start-up level and make something big.

With this platform students will get real time feedback and reviews on their project , which will help them improve it and make it better .

This will encourage more and more students to move towards job creation , rather than job finding . This platform would come out to be helpful for a lot of students in college and higher levels .

REFERENCES

- linkedIn.com
- medium.com
- w3schools.com
- researchgate.com
- youtube.com
- materialise.com
- material.io
- dribblec.com