Internet and Web Programming

Course Code: CSE3002

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

QUOVIT

An extended version of Quora for college

Team Members	Mounvi Podapati (19BCE0396)
Slot	L25 + L26 (G2)
Faculty	Prof. Lydia Jane

INTRODUCTION:

QuoVIT is a social media platform for VIT students. Offering various facilities which the students belonging to an educational institution need. It serves as an all in one. While providing multi-faceted facilities ranging from the traditional social media posts to more college/VIT specific functionalities like faculty revies, doubt clarifications forum and more facilities. All of this comes under the umbrella of one website which caters to the needs of an enormous educational institution like VIT. To make sure not to apply the one-fits all model to digital networking platforms and have thus chosen to take it upon to provide institution specific functionalities on this portal. Quovit makes the lives of a student easier, by giving them one place for all their needs related to college. Be it discussions, socializing, information gathering; QuoVIT has it all. It aims to improve the quality of life and education by providing a vital support system to the average student in order to empower them to make well informed decisions and have better control over their fate.

AIM:

My Aim is to develop an application that acts as a single platform catering all the needs of students belonging a big educational institution like VIT. To come up with a system that specifically acts as a social media application exclusively for students of VIT along with providing other functionalities like a Question Bank system, Faculty reviews section, an Ideas sharing platform and a module to allow students to anonymously share their confessions.

OBJECTIVE:

Quovit's main objective is to act as a single reliable and sufficient application instead of relying on multiple different applications for daily student related requirements. To come up with a MERN stack application for the same.

TECH-STACK:

MongoDB : A NoSQL based Document Database

Express: Node.js web framework

React: Client side Javascript framework

NodeJS: Javascript based Web Server

M E R N

MODULES:

1. Posts:

Typical to what posts on popular social media platforms look like, the posts on QuoVIT are aimed at increasing the social networking aspect of a student's internal college life. The posts can be made in the form of text or . This empowers a user to easily make the best use of their time and understand what his or her mates in college are doing. This would help in nurturing a professional yet very social and inclusive environment and culture in the institution.

2. Anonymous Confessions:

This is one for the public. A module where anyone can say something which holds a lot of value and truth but don't want to go public with your identity disclosed. This page is for those people looking to be heard but also avoid the heat. It offers a platform for users to convey their ideas/issues without disclosing their identities and thus keeping the heat off their backs. The existence of such a device simply calls for increased transparency withing the institution which would help it to flourish in the future.

3. Question bank:

This module is specifically designed to allow students to view and download question papers over the years, exam category wise so that they can get an idea of the pattern, type and level of the questions asked and the marking scheme and hence help students in testing their knowledge and practise various questions. Students will also be allowed to upload and help other students with the same in the future.

4. Idea sharing:

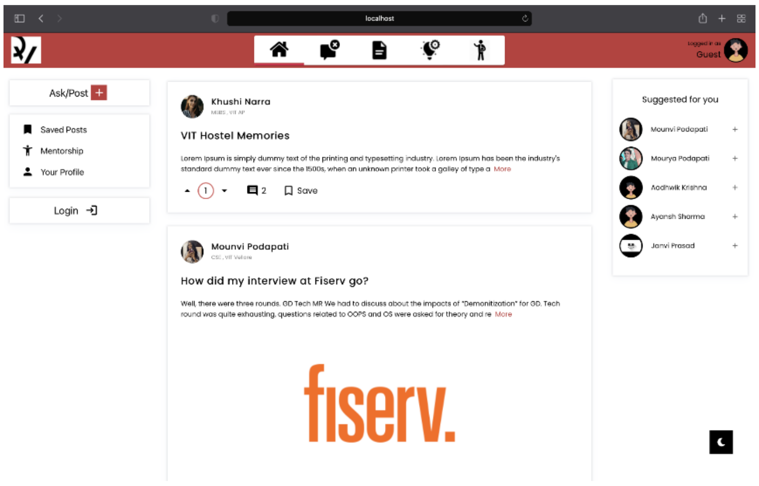
A module to allow students to share or putting out innovative ideas into the world just to see the response it gets? In that case, the ideas sharing portal is the place to be. Here the user can share his/her ideas and make sure it gets some audience since it automatically goes onto the institution's digital platform. This enables other users to make the choice of either helping in the solidification of the idea or just upvoting or supporting its progress.

5. Faculty Review:

A module to allow students to select the right faculty during FFCS. This will help students to make the correct and well informed decision about taking the righ faculty and the right subject to help you in you adventurous and fruitful journey of your degree completion. Find overall rating on multiple faculties and make the best informed and logical choice.

SCREENSHOTS:

Landing Page:


localhost

Logged in as Guest

Ask/Post +

Saved Posts
Mentorship
Your Profile

Login

Khushi Narra
MBBS , VIT AP

VIT Hostel Memories

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type a More

1 2 Save

Mounvi Podapati
CSE , VIT Vellore

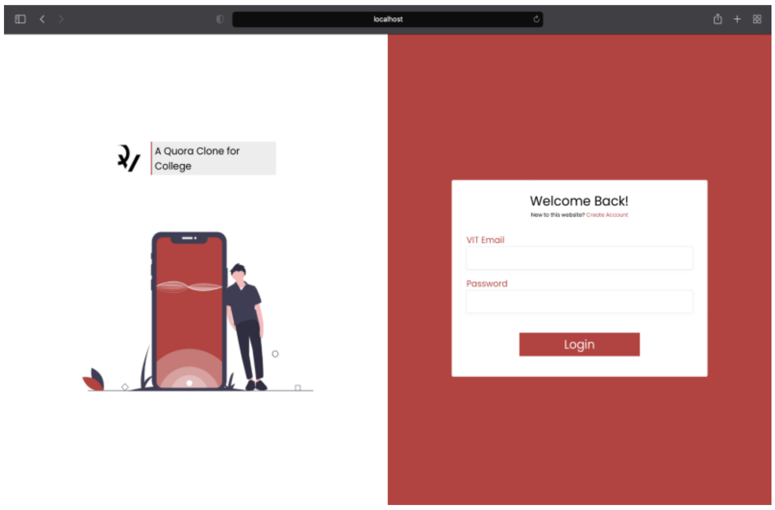
How did my interview at Fiserv go?

Well, there were three rounds. GD Tech MR We had to discuss about the impacts of "Demonitization" for GD. Tech round was quite exhausting, questions related to OOPS and OS were asked for theory and re More

Suggested for you

Mounvi Podapati +
Mourya Podapati +
Aadhwik Krishna +
Ayansh Sharma +
Janvi Prasad +

Clicking on the Upvote button on a post. (Redirected to login as the guest users are not allowed to react on any post, idea, confession, downloaded question papers or rate any faculties or follow and unfollow users)


localhost

A Quora Clone for College

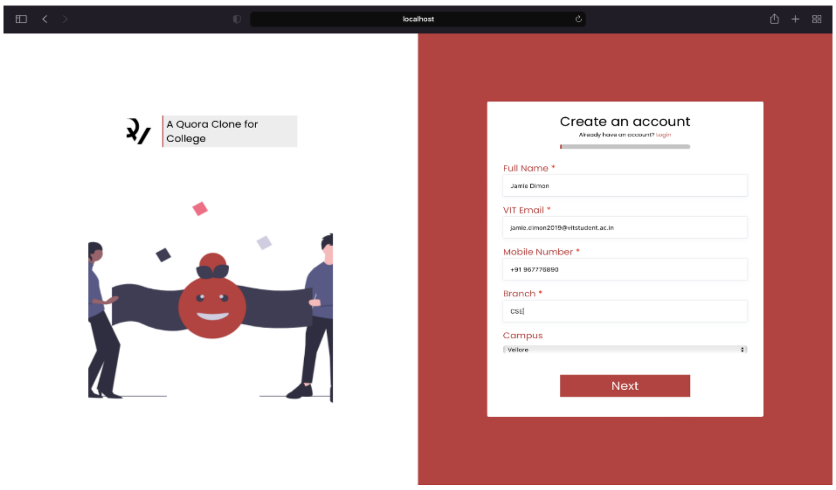
Welcome Back!
New to this website? Create Account

VIT Email

Password

Login

Clicking on Create Account and creating a new user.


localhost

A Quora Clone for College

Create an account
Already have an account? Login

Full Name *
Jamie Dimon

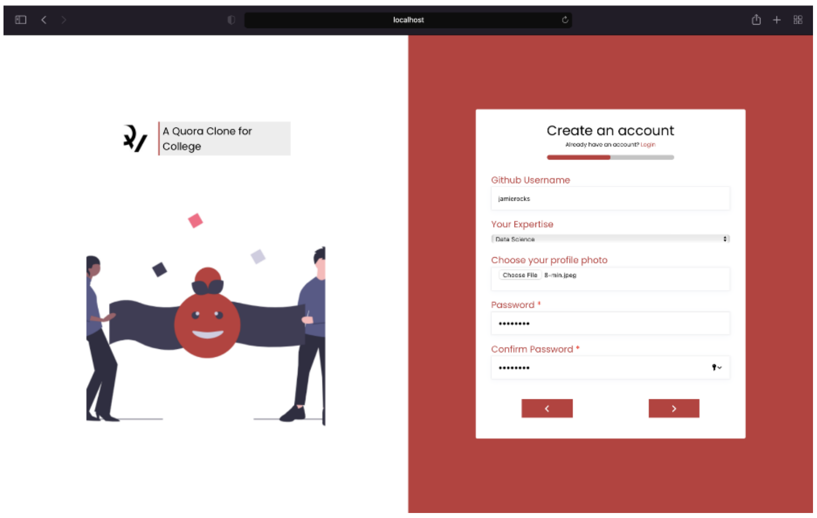
VIT Email *
jamie.dimon2019@vitstudent.ac.in

Mobile Number *
+91 967776890

Branch *
CSE

Campus
Vellore

Next



Create an account

Already have an account? Login

Github Username: jamierocks

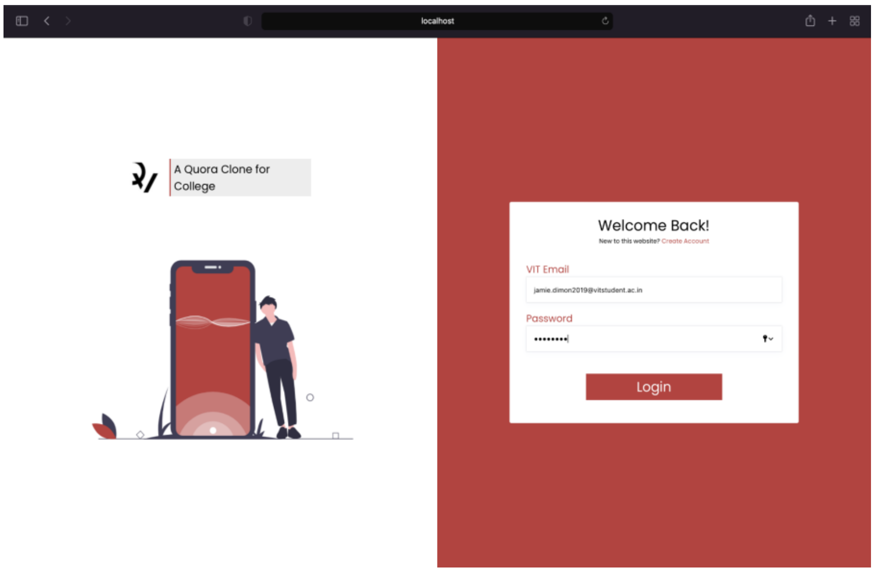
Your Expertise: Data Science

Choose your profile photo: Choose File 8-min.jpeg

Password *: ••••••••

Confirm Password *: ••••••••

On successful creation of account, you will be redirected to the Login page



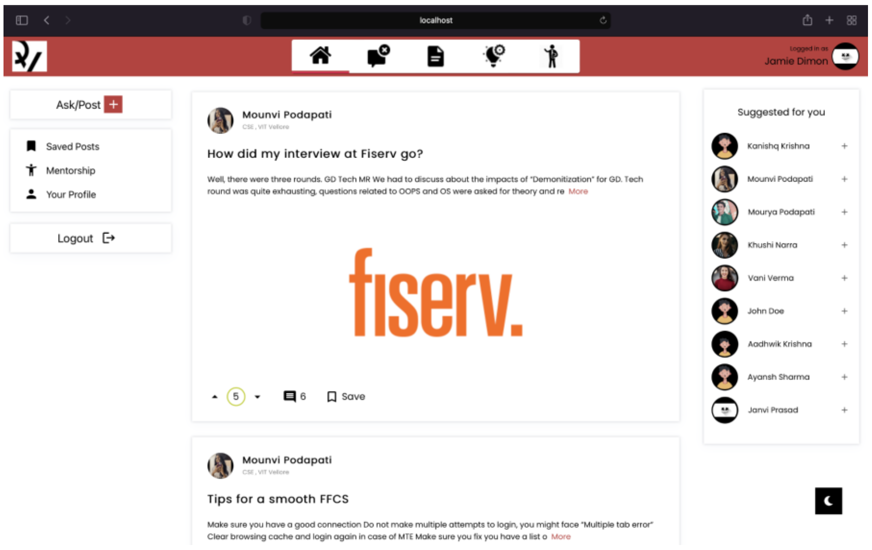
Welcome Back!

New to this website? Create Account

VIT Email: jamie.dimon2019@vitstudent.ac.in

Password: ••••••••

Logged in user:



Mounvi Podapati
CSE , VIT Vellore

How did my interview at Fiserv go?

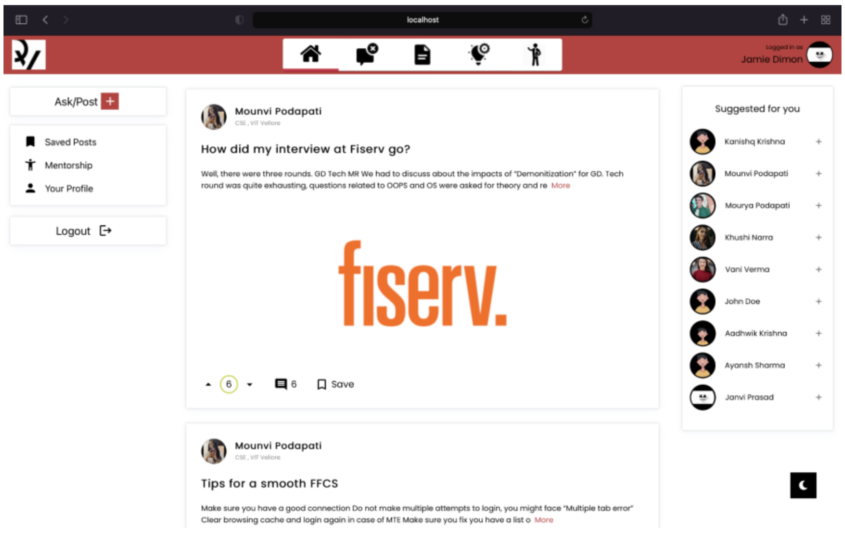
Well, there were three rounds. GD Tech MR We had to discuss about the impacts of "Demonitization" for GD. Tech round was quite exhausting, questions related to OOPS and OS were asked for theory and re More

Mounvi Podapati
CSE , VIT Vellore

Tips for a smooth FFCS

Make sure you have a good connection Do not make multiple attempts to login, you might face "Multiple tab error" Clear browsing cache and login again in case of MTE Make sure you fix you have a list o More

Upvoting on the first post:



localhost

Logged in as Jamie Dimon

Ask/Post +

Saved Posts
Mentorship
Your Profile

Logout

Mounvi Podapati
CSE , VIT Vellore

How did my interview at Fiserv go?

Well, there were three rounds. GD Tech MR We had to discuss about the impacts of "Demonitization" for GD. Tech round was quite exhausting, questions related to OOPS and OS were asked for theory and re More

fiserv.

6 6 Save

Mounvi Podapati
CSE , VIT Vellore

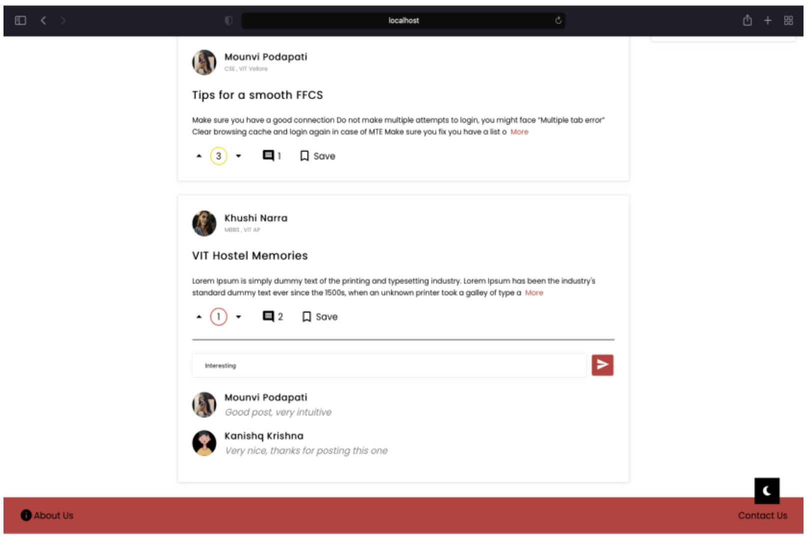
Tips for a smooth FFCS

Make sure you have a good connection Do not make multiple attempts to login, you might face "Multiple tab error" Clear browsing cache and login again in case of MTE Make sure you fix you have a list o More

Suggested for you

Kanishq Krishna +
Mounvi Podapati +
Mourya Podapati +
Khushi Narra +
Vani Verma +
John Doe +
Aadhwik Krishna +
Ayansh Sharma +
Janvi Prasad +

Commenting on the third post:



localhost

Mounvi Podapati
CSE , VIT Vellore

Tips for a smooth FFCS

Make sure you have a good connection Do not make multiple attempts to login, you might face "Multiple tab error" Clear browsing cache and login again in case of MTE Make sure you fix you have a list o More

3 1 Save

Khushi Narra
MBBS , VIT AP

VIT Hostel Memories

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type a More

1 2 Save

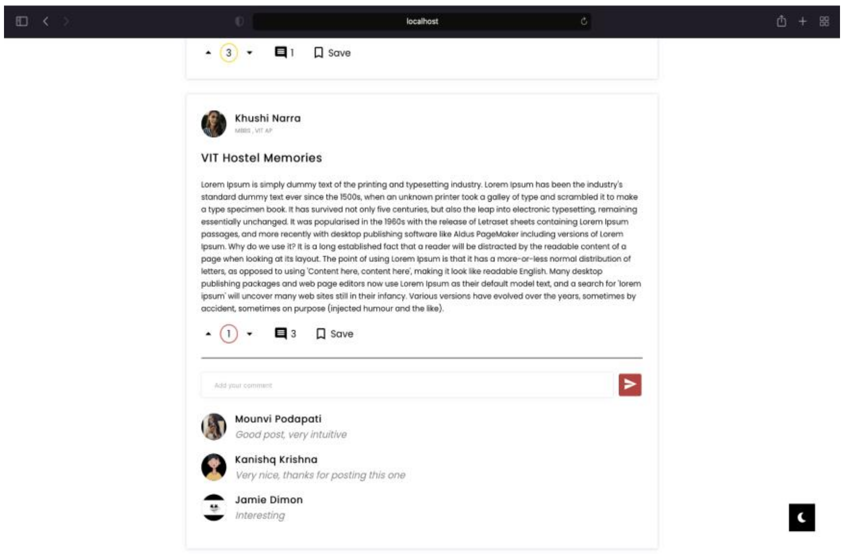
Interesting

Mounvi Podapati
Good post, very intuitive

Kanishq Krishna
Very nice, thanks for posting this one

About Us

Contact Us



localhost

3 1 Save

Khushi Narra
MBBS , VIT AP

VIT Hostel Memories

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

1 3 Save

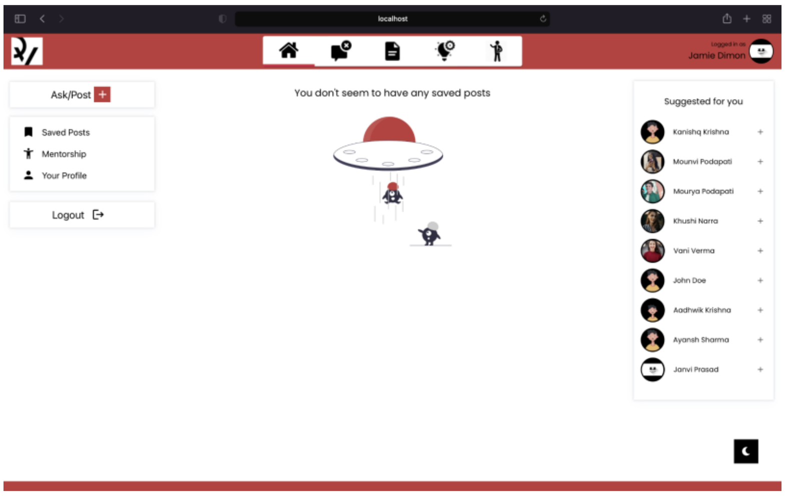
Add your comment

Mounvi Podapati
Good post, very intuitive

Kanishq Krishna
Very nice, thanks for posting this one

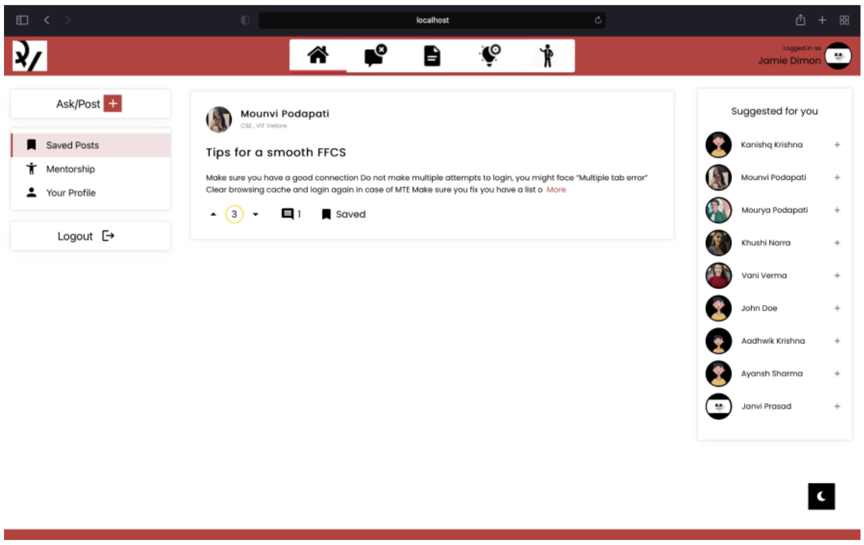
Jamie Dimon
Interesting

Saved posts page:



You don't seem to have any saved posts

Going back to landing and saving the second post:



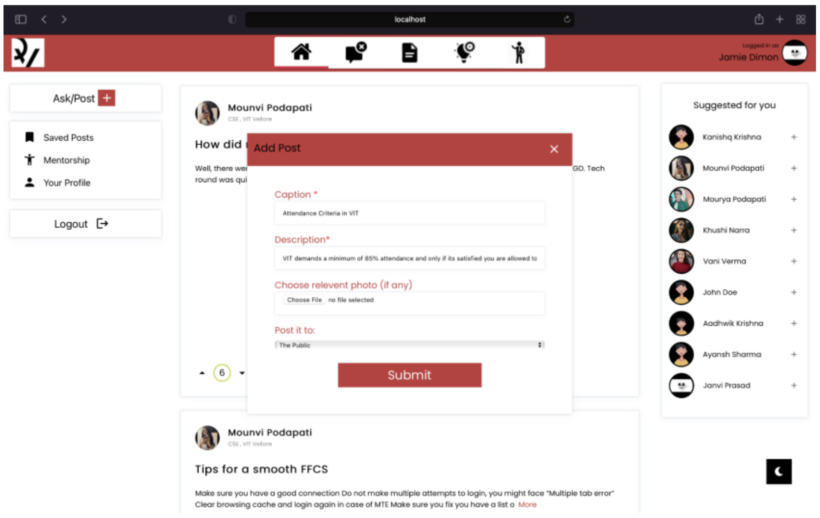
Mounvi Podapati
CSE , VIT Vellore

Tips for a smooth FFCS

Make sure you have a good connection Do not make multiple attempts to login, you might face "Multiple tab error" Clear browsing cache and login again in case of MTE Make sure you fix you have a list o More

3 · 1 · Saved

Creating a new post and posting to your followers only:



Mounvi Podapati
CSE , VIT Vellore

How did

Well, there wer ... GD. Tech round was qui

6

Add Post

Caption *
Attendance Criteria in VIT

Description*
VIT demands a minimum of 85% attendance and only if its satisfied you are allowed to

Choose relevent photo (if any)
Choose File no file selected

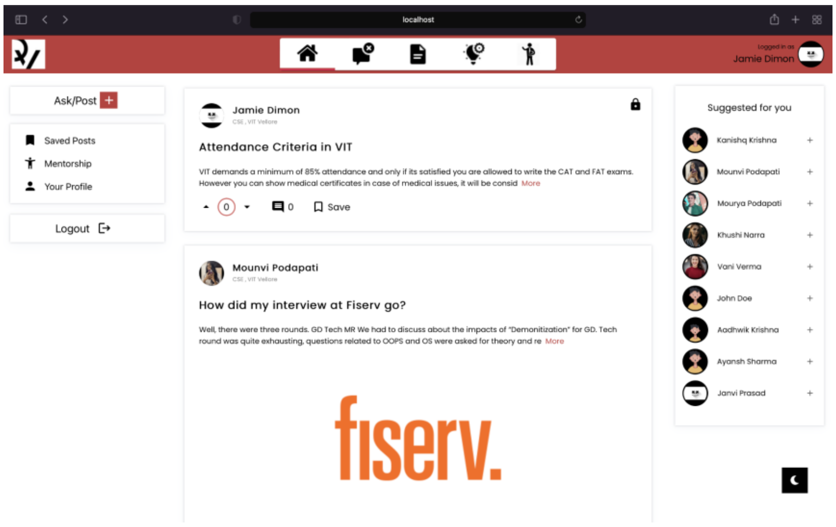
Post it to:
The Public

Submit

Mounvi Podapati
CSE , VIT Vellore

Tips for a smooth FFCS

Make sure you have a good connection Do not make multiple attempts to login, you might face "Multiple tab error" Clear browsing cache and login again in case of MTE Make sure you fix you have a list o More


localhost

Logged in as Jamie Dimon

Ask/Post +

Saved Posts
Mentorship
Your Profile

Logout

Jamie Dimon
CSE , VIT Vellore

Attendance Criteria in VIT

VIT demands a minimum of 85% attendance and only if its satisfied you are allowed to write the CAT and FAT exams. However you can show medical certificates in case of medical issues, it will be consid More

0 0 Save

Mounvi Podapati
CSE , VIT Vellore

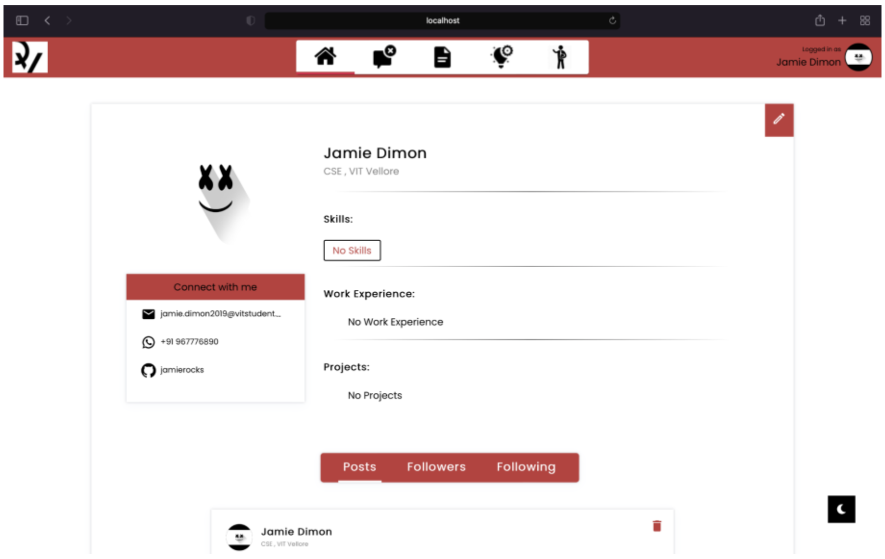
How did my interview at Fiserv go?

Well, there were three rounds. GD Tech MR We had to discuss about the impacts of "Demonitization" for GD. Tech round was quite exhausting, questions related to OOPS and OS were asked for theory and re More

Suggested for you

Kanishq Krishna +
Mounvi Podapati +
Mourya Podapati +
Khushi Narra +
Vani Verma +
John Doe +
Aadhwik Krishna +
Ayansh Sharma +
Janvi Prasad +

Clicking on Your Profile button


localhost

Logged in as Jamie Dimon

Jamie Dimon
CSE , VIT Vellore

Skills:

No Skills

Work Experience:

No Work Experience

Projects:

No Projects

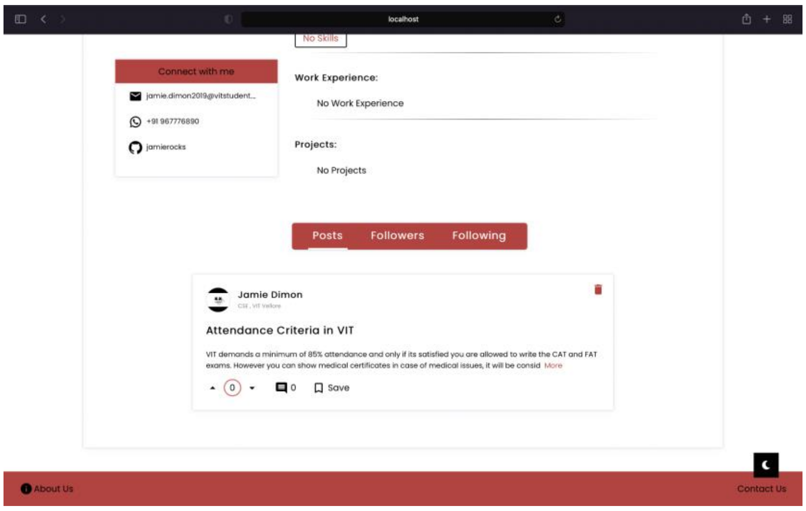
Connect with me

jamie.dimon2019@vitstudent...
+91 967776890
jamierocks

Posts Followers Following

Jamie Dimon
CSE , VIT Vellore

Deleting the post we made previously


localhost

No Skills

Work Experience:

No Work Experience

Projects:

No Projects

Connect with me

jamie.dimon2019@vitstudent...
+91 967776890
jamierocks

Posts Followers Following

Jamie Dimon
CSE , VIT Vellore

Attendance Criteria in VIT

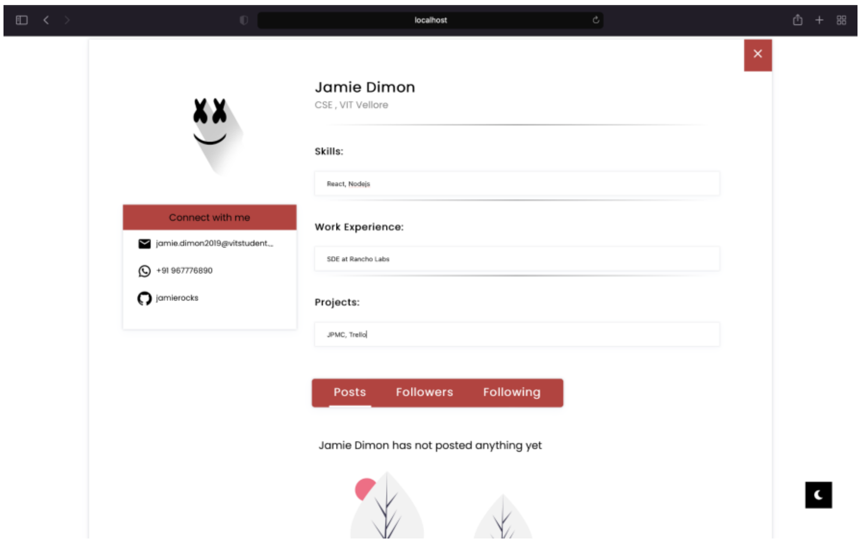
VIT demands a minimum of 85% attendance and only if its satisfied you are allowed to write the CAT and FAT exams. However you can show medical certificates in case of medical issues, it will be consid More

0 0 Save

About Us

Contact Us

Editing to add new skills:



Jamie Dimon
CSE , VIT Vellore

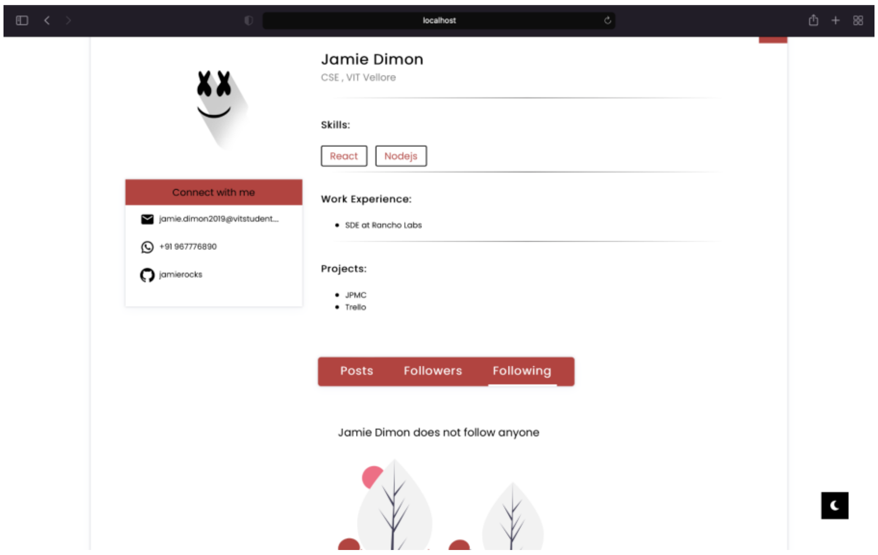
Skills:
React, Nodejs

Work Experience:
SDE at Rancho Labs

Projects:
JPMC, Trello

Connect with me
jamie.dimon2019@vitstudent...
+91 967776890
jamierocks

Jamie Dimon has not posted anything yet



Jamie Dimon
CSE , VIT Vellore

Skills:
React Nodejs

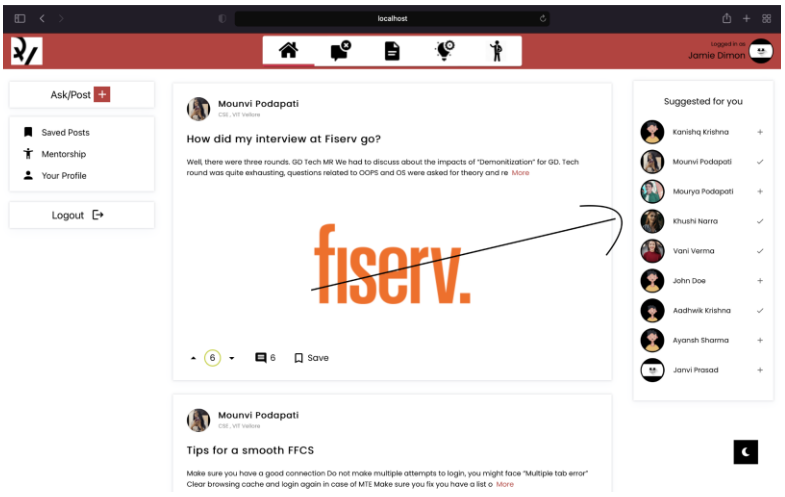
Work Experience:
- SDE at Rancho Labs

Projects:
- JPMC
- Trello

Connect with me
jamie.dimon2019@vitstudent...
+91 967776890
jamierocks

Jamie Dimon does not follow anyone

Following users shown in suggestions:



Mounvi Podapati
CSE , VIT Vellore

How did my interview at Fiserv go?

Well, there were three rounds. GD Tech MR We had to discuss about the impacts of "Demonitization" for GD. Tech round was quite exhausting, questions related to OOPS and OS were asked for theory and re More

Mounvi Podapati
CSE , VIT Vellore

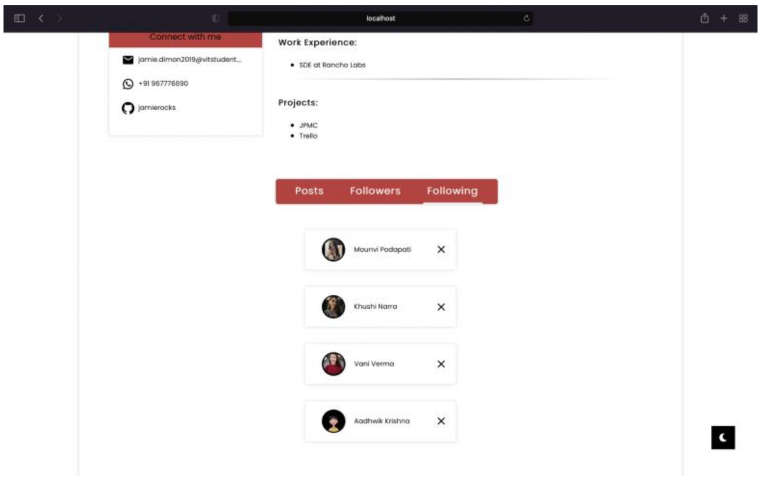
Tips for a smooth FFCS

Make sure you have a good connection Do not make multiple attempts to login, you might face "Multiple tab error" Clear browsing cache and login again in case of MTE Make sure you fix you have a list o More

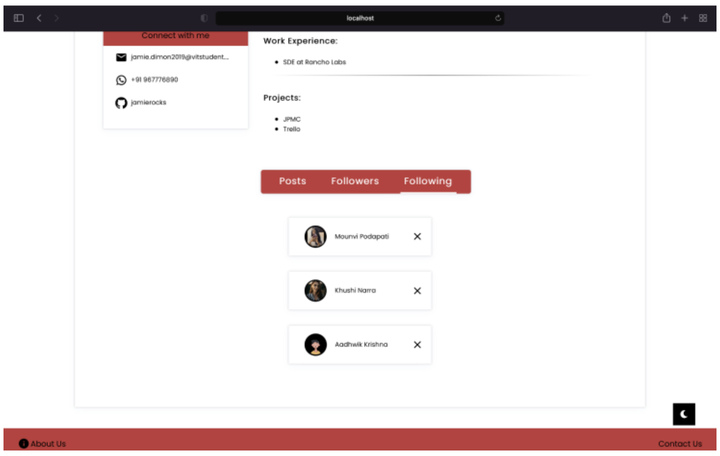
Suggested for you

Kanishq Krishna	+
Mounvi Podapati	✓
Mourya Podapati	+
Khushi Narra	✓
Vani Verma	✓
John Doe	+
Aadhwik Krishna	✓
Ayansh Sharma	+
Janvi Prasad	+

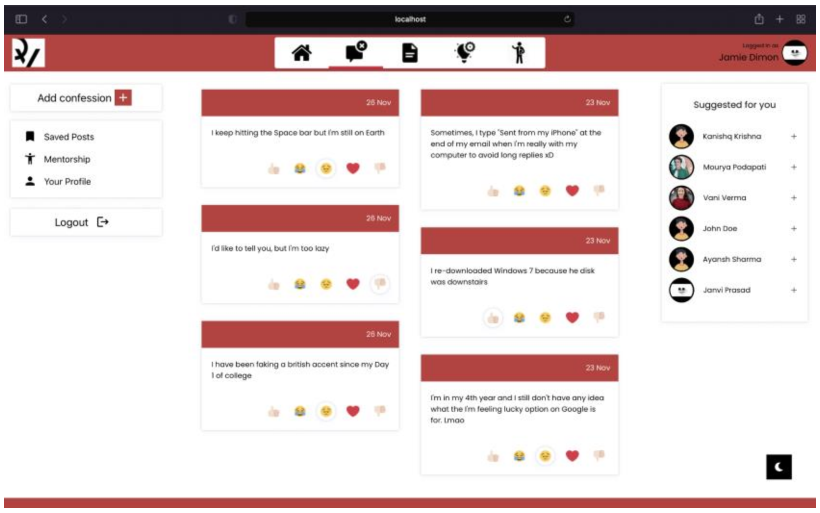
It will now be reflected in the profile page under following:



Users can be unfollowed by clicking on cross symbol: Unfollowing user Vani Verma



Anonymous confessions page: Reacting to multiple confessions:



I keep hitting the Space bar but I'm still on Earth

I'd like to tell you, but I'm too lazy

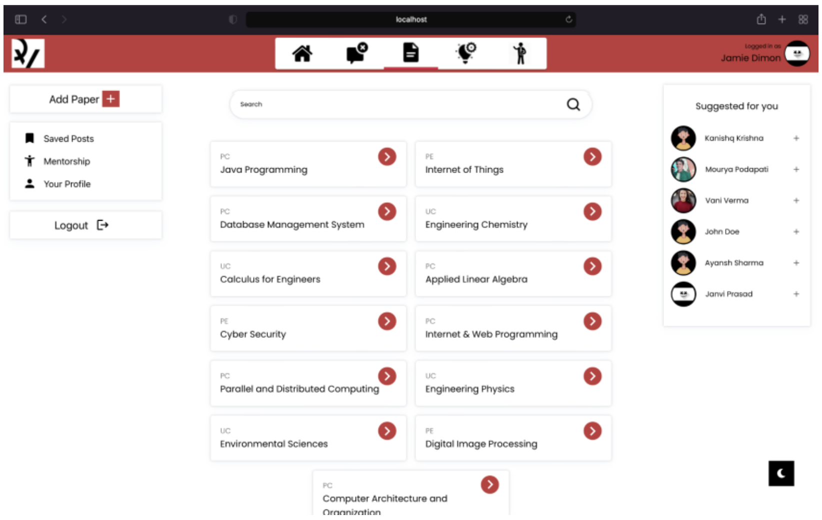
I have been faking a british accent since my Day 1 of college

Sometimes, I type "Sent from my iPhone" at the end of my email when I'm really with my computer to avoid long replies xD

I re-downloaded Windows 7 because he disk was downstairs

I'm in my 4th year and I still don't have any idea what the I'm feeling lucky option on Google is for. lmao

Question Bank:



localhost

Logged in as Jamie Dimon

Add Paper +

Saved Posts
Mentorship
Your Profile

Logout

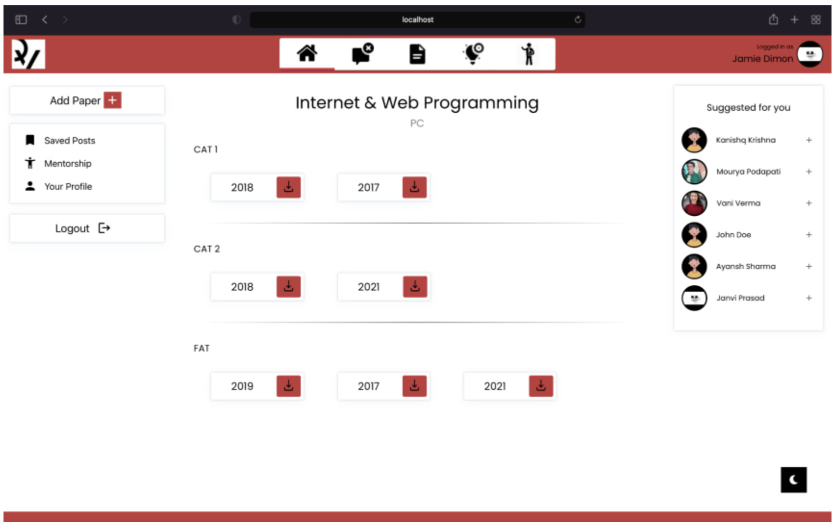
Search

PC Java Programming	PE Internet of Things
PC Database Management System	UC Engineering Chemistry
UC Calculus for Engineers	PC Applied Linear Algebra
PE Cyber Security	PC Internet & Web Programming
PC Parallel and Distributed Computing	UC Engineering Physics
UC Environmental Sciences	PE Digital Image Processing
PC Computer Architecture and Organization	

Suggested for you

Kanishq Krishna +
Mourya Podapati +
Vani Verma +
John Doe +
Ayansh Sharma +
Janvi Prasad +

Clicking on Internet and Web Programming:



localhost

Logged in as Jamie Dimon

Add Paper +

Saved Posts
Mentorship
Your Profile

Logout

Internet & Web Programming

PC

CAT 1

2018 2017

CAT 2

2018 2021

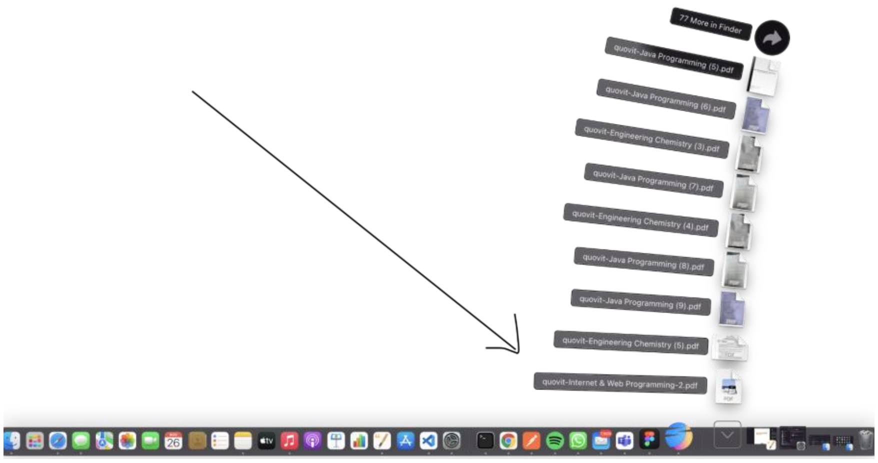
FAT

2019 2017 2021

Suggested for you

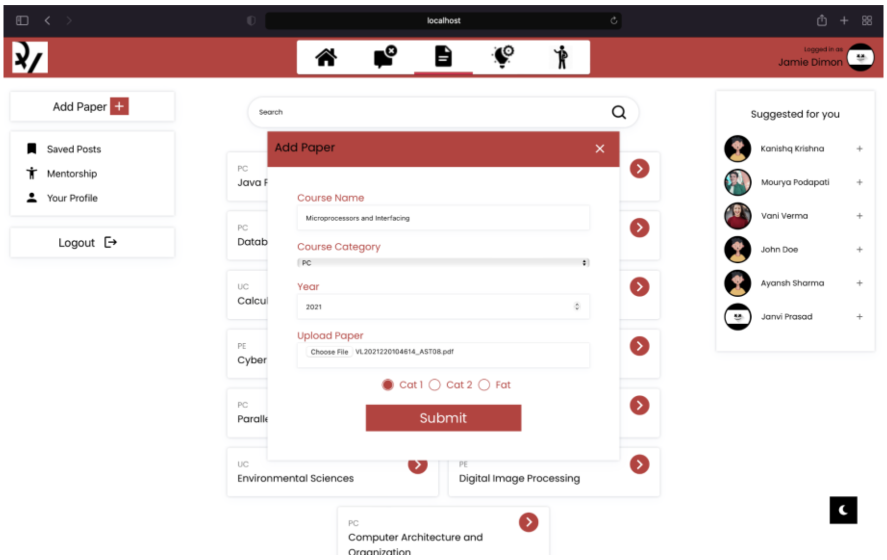
Kanishq Krishna +
Mourya Podapati +
Vani Verma +
John Doe +
Ayansh Sharma +
Janvi Prasad +

Question papers can be downloaded on clicking on the download icon:



77 More in Finder
quovit-Java Programming (5).pdf
quovit-Java Programming (6).pdf
quovit-Engineering Chemistry (3).pdf
quovit-Java Programming (7).pdf
quovit-Engineering Chemistry (4).pdf
quovit-Java Programming (8).pdf
quovit-Java Programming (9).pdf
quovit-Engineering Chemistry (5).pdf
quovit-Internet & Web Programming-2.pdf

Adding a new course:



Add Paper

Course Name: Microprocessors and Interfacing

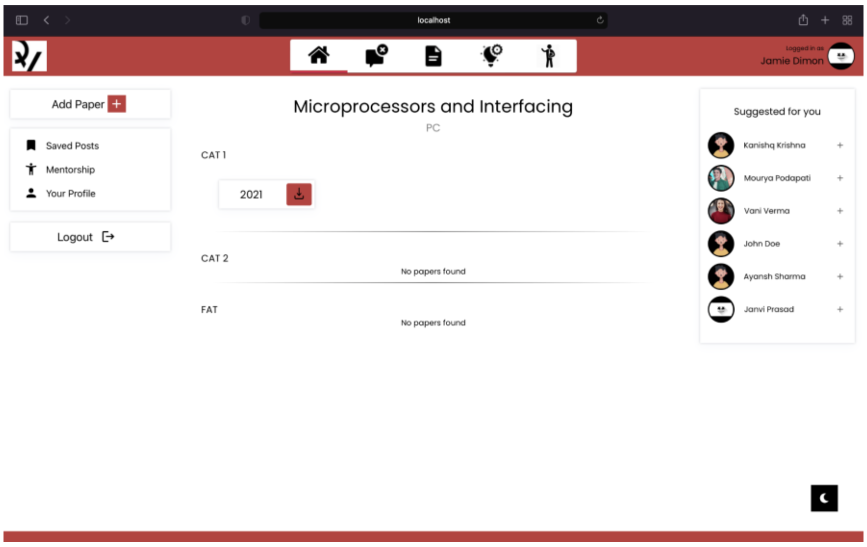
Course Category: PC

Year: 2021

Upload Paper: Choose File VL2021220104614_AST08.pdf

[✓ Cat 1] Cat 2 Fat

Submit



Microprocessors and Interfacing

PC

CAT 1

2021

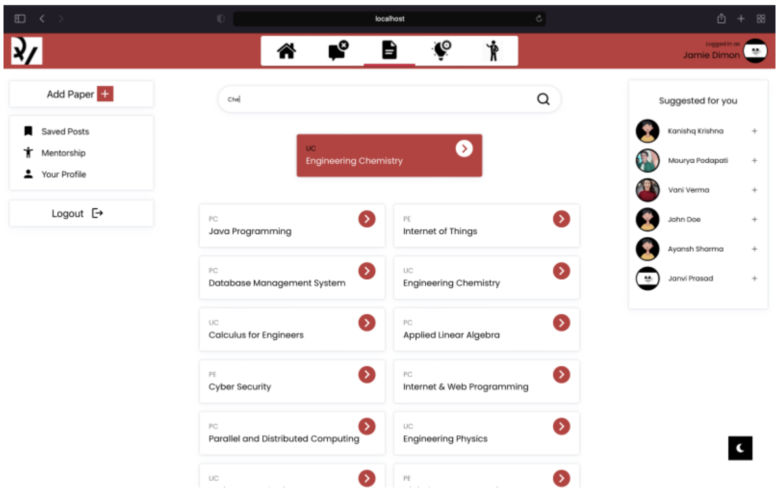
CAT 2

No papers found

FAT

No papers found

Searching for a course:



Che

UC
Engineering Chemistry

PC
Java Programming

PE
Internet of Things

PC
Database Management System

UC
Engineering Chemistry

UC
Calculus for Engineers

PC
Applied Linear Algebra

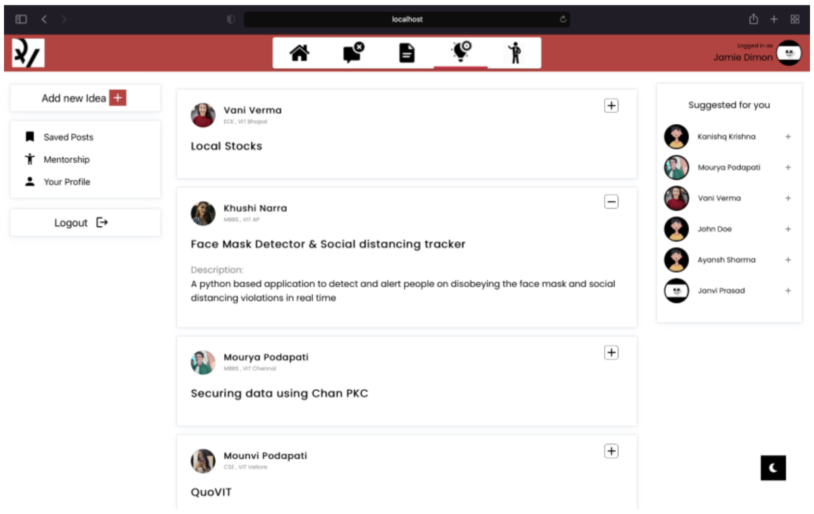
PE
Cyber Security

PC
Internet & Web Programming

PC
Parallel and Distributed Computing

UC
Engineering Physics

Ideas sharing platform: Description can be viewed by clicking on the + icon:



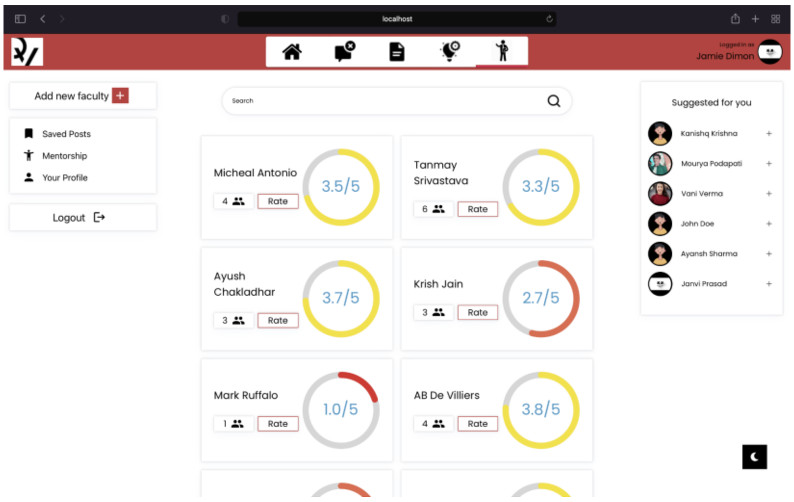
Vani Verma
ECE, VIT Bhopal
Local Stocks

Khushi Narra
MBBS, VIT AP
Face Mask Detector & Social distancing tracker
Description:
A python based application to detect and alert people on disobeying the face mask and social distancing violations in real time

Mourya Podapati
MBBS, VIT Chennai
Securing data using Chan PKC

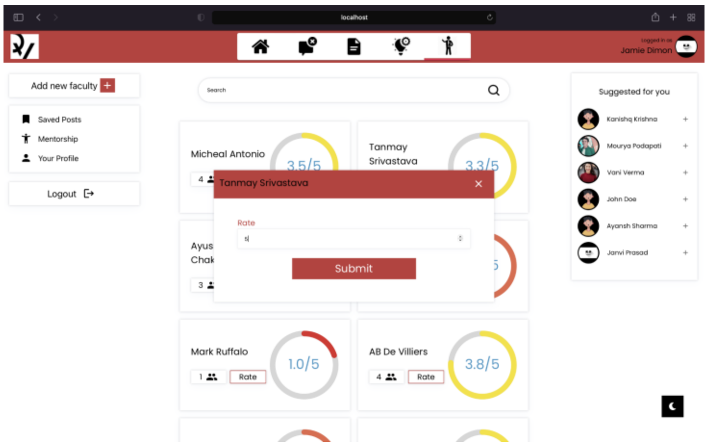
Mounvi Podapati
CSE, VIT Vellore
QuoVIT

Faculty Reviews Page:



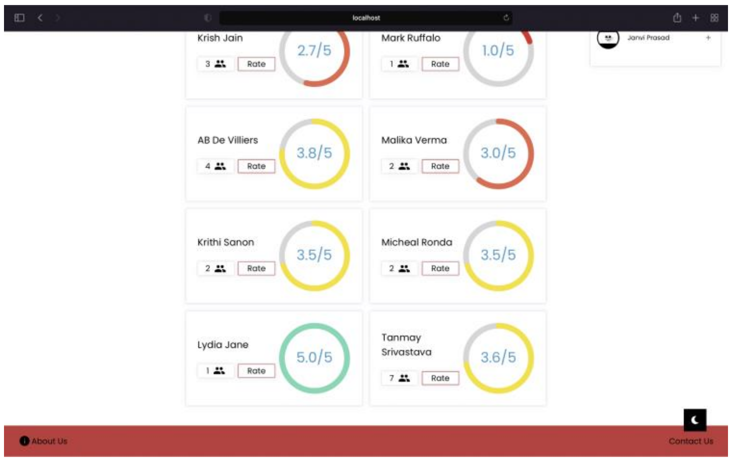
Micheal Antonio	4	3.5/5
Tanmay Srivastava	6	3.3/5
Ayush Chakladhar	3	3.7/5
Krish Jain	3	2.7/5
Mark Ruffalo	1	1.0/5
AB De Villiers	4	3.8/5

Rating a faculty: Tanmay Srivastava



Tanmay Srivastava
Rate
5
Submit

Faculty Votes and overall rating has been updated


localhost

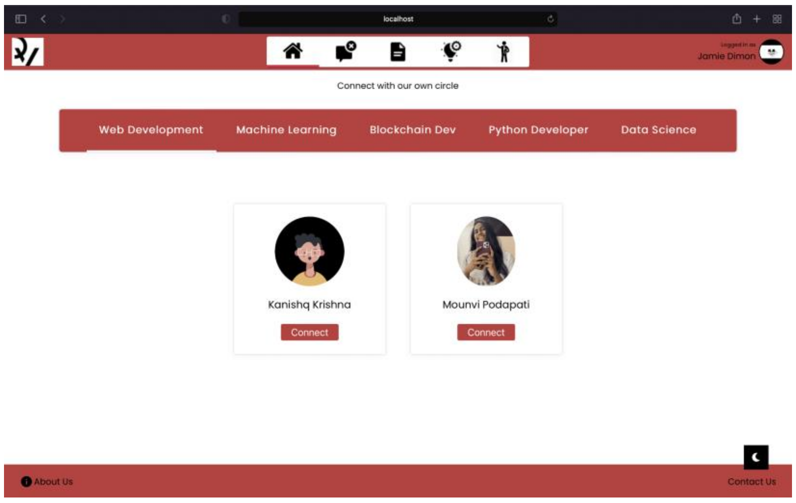
Krish Jain 3 Rate 2.7/5
Mark Ruffalo 1 Rate 1.0/5
AB De Villiers 4 Rate 3.8/5
Malika Verma 2 Rate 3.0/5
Krithi Sanon 2 Rate 3.5/5
Micheal Ronda 2 Rate 3.5/5
Lydia Jane 1 Rate 5.0/5
Tanmay Srivastava 7 Rate 3.6/5

Janvi Prasad

About Us Contact Us

Clicking on mentorship button in the sidebar:

Users are filtered according to different domains so that beginners can connect to them


localhost

Logged in as Jamie Dimon

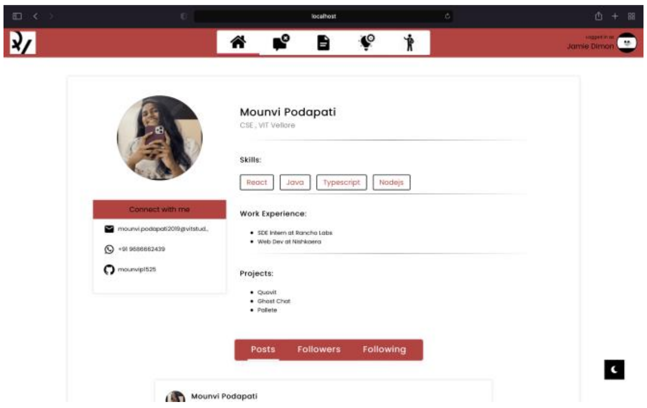
Connect with our own circle

Web Development | Machine Learning | Blockchain Dev | Python Developer | Data Science

Kanishq Krishna Connect
Mounvi Podapati Connect

About Us Contact Us

Clicking on connect button: will be redirected to their profile page:


localhost

Logged in as Jamie Dimon

Mounvi Podapati
CSE , VIT Vellore

Skills:
React Java Typescript Nodejs

Work Experience:
- SDE Intern at Rancho Labs
- Web Dev at Nishkaera

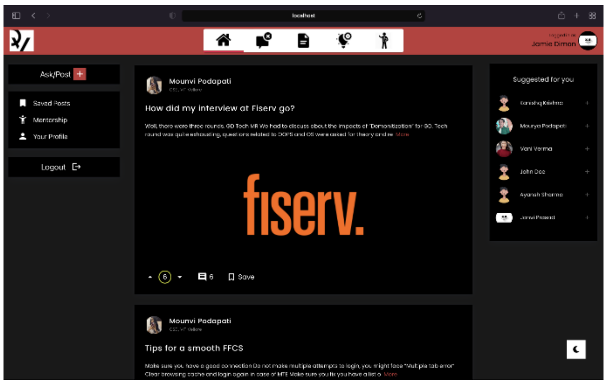
Projects:
- Quovit
- Ghost Chat
- Pallete

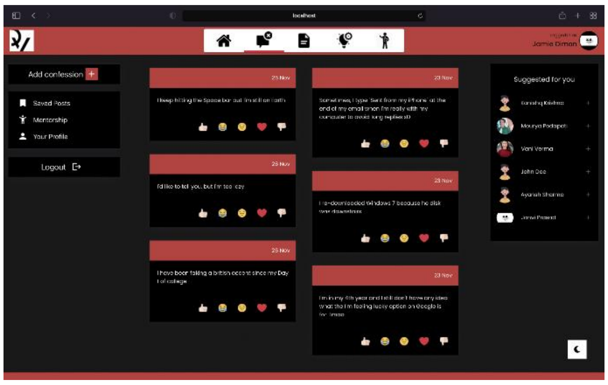
Connect with me
mounvi.podapati2019@vitstud..
+91 9686662439
mounvipl525

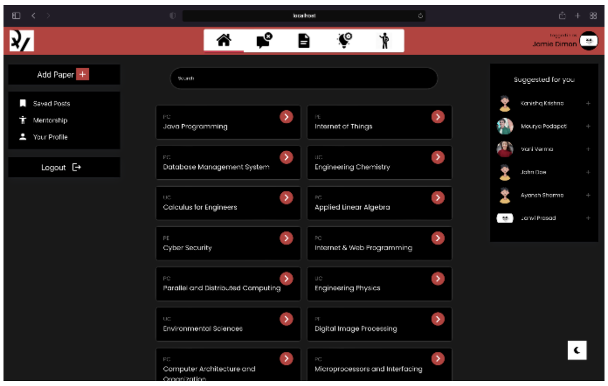
Posts Followers Following

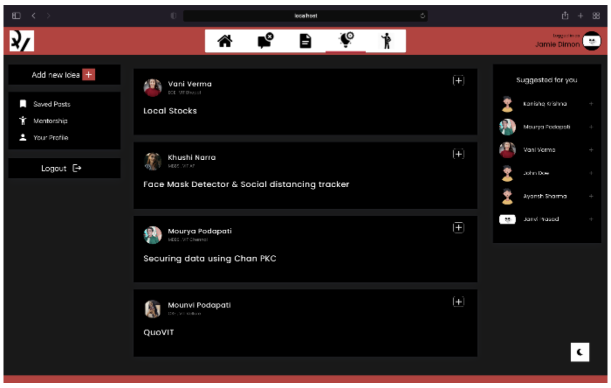
Mounvi Podapati

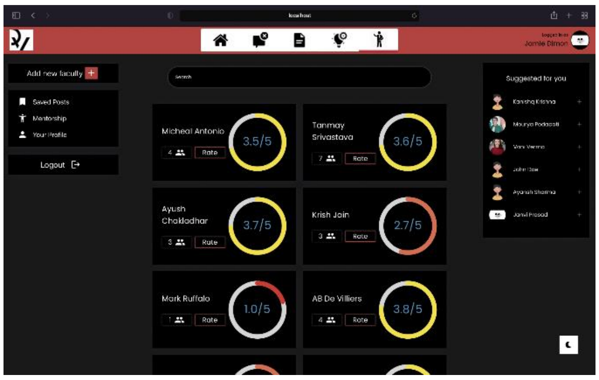
Dark mode added as well:



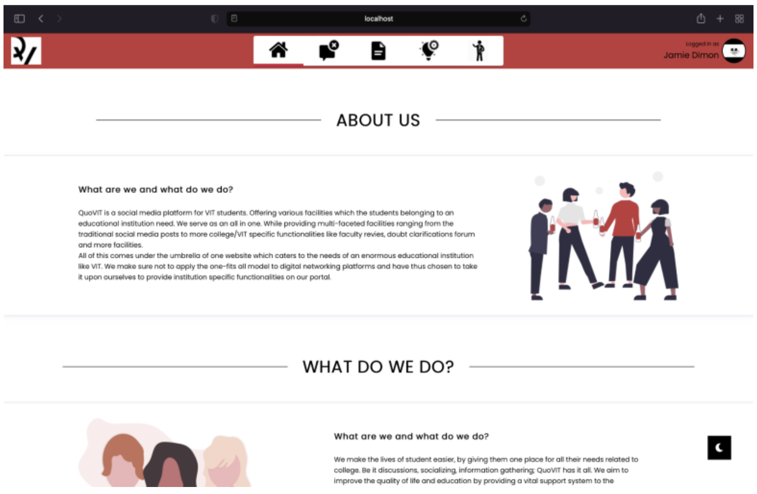








About us page:



ABOUT US

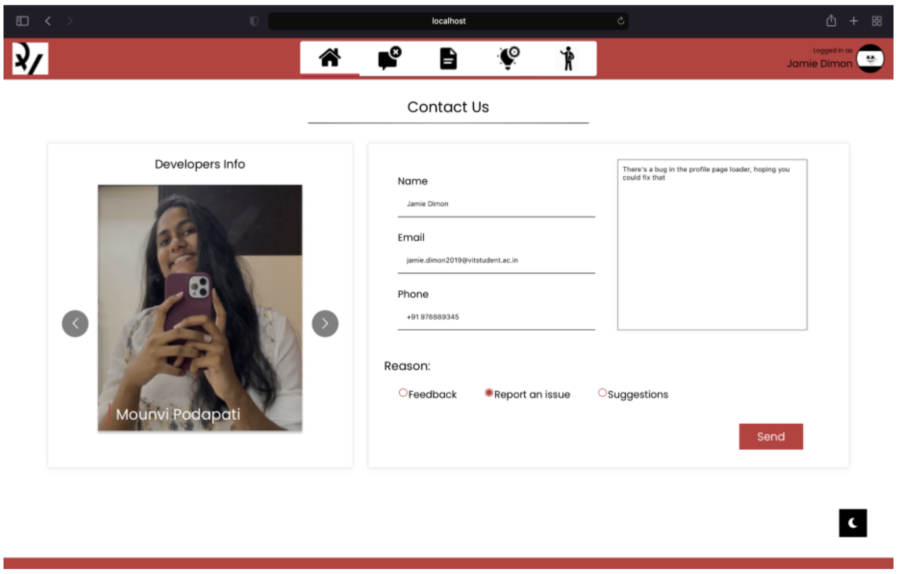
What are we and what do we do?

QuoVIT is a social media platform for VIT students. Offering various facilities which the students belonging to an educational institution need. We serve as an all in one. While providing multi-faceted facilities ranging from the traditional social media posts to more college/VIT specific functionalities like faculty revies, doubt clarifications forum and more facilities.
All of this comes under the umbrella of one website which caters to the needs of an enormous educational institution like VIT. We make sure not to apply the one-fits all model to digital networking platforms and have thus chosen to take it upon ourselves to provide institution specific functionalities on our portal.

WHAT DO WE DO?

What are we and what do we do?

We make the lives of student easier, by giving them one place for all their needs related to college. Be it discussions, socializing, information gathering; QuoVIT has it all. We aim to improve the quality of life and education by providing a vital support system to the



Contact Us

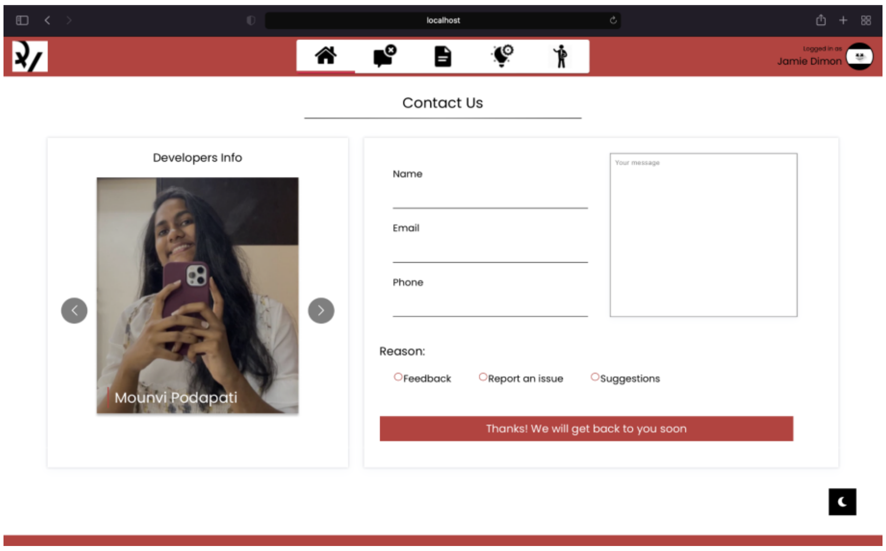
Name: Jamie Dimon

Email: jamie.dimon2019@vitstudent.ac.in

Phone: +91 978889345

There's a bug in the profile page loader, hoping you could fix that

Reason: Feedback [✓ Report an issue] Suggestions



Contact Us

Name

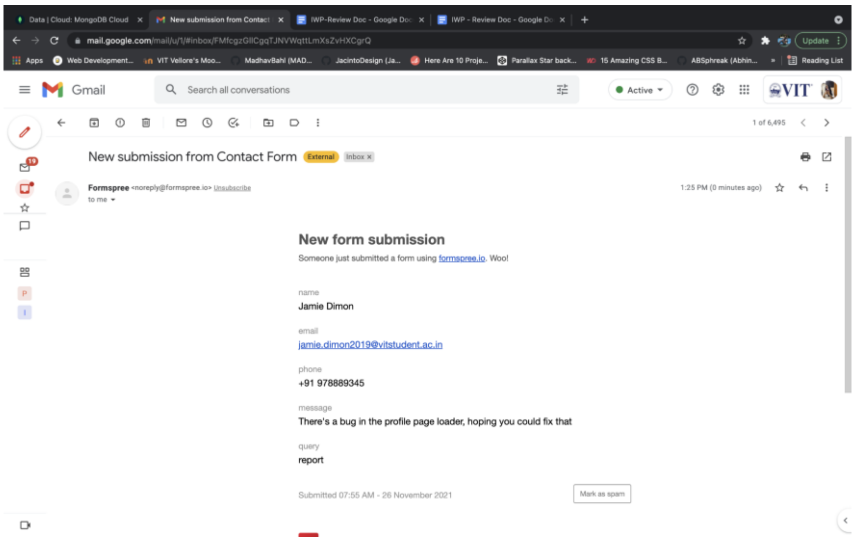
Email

Phone

Reason: Feedback Report an issue Suggestions

Thanks! We will get back to you soon

This message will be sent as an email to the admin



New submission from Contact Form

Formspree <noreply@formspree.io> Unsubscribe
to me

1:25 PM (0 minutes ago)

New form submission

Someone just submitted a form using formspree.io. Woo!

name
Jamie Dimon

email
jamie.dimon2019@vitstudent.ac.in

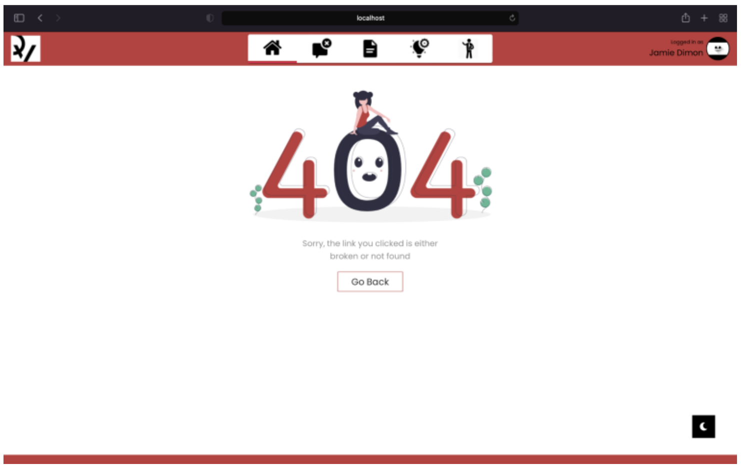
phone
+91 978889345

message
There's a bug in the profile page loader, hoping you could fix that

query
report

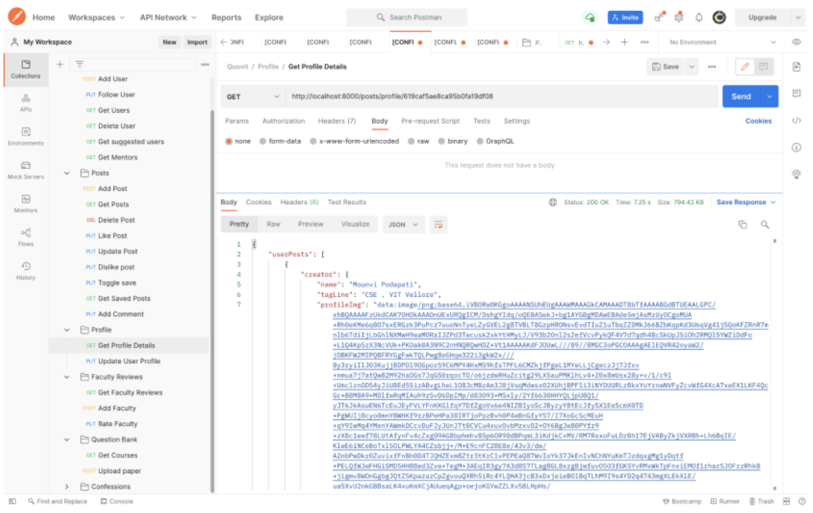
Submitted 07:55 AM - 26 November 2021

Trying to access a link that doesn't exist:



Sorry, the link you clicked is either broken or not found

Postman Routes Testing:



POST Add User
PUT Follow User
GET Get Users
GET Delete User
GET Get suggested users
GET Get Mentors
Posts
POST Add Post
GET Get Posts
DEL Delete Post
PUT Like Post
PUT Update Post
PUT Dislike post
PUT Toggle save
GET Get Saved Posts
PUT Add Comment
Profile
GET Get Profile Details
PUT Update User Profile
Faculty Reviews
GET Get Faculty Reviews
POST Add Faculty
PUT Rate Faculty
Question Bank
GET Get Courses
POST Upload paper
Confessions

Quovit / Profile / Get Profile Details

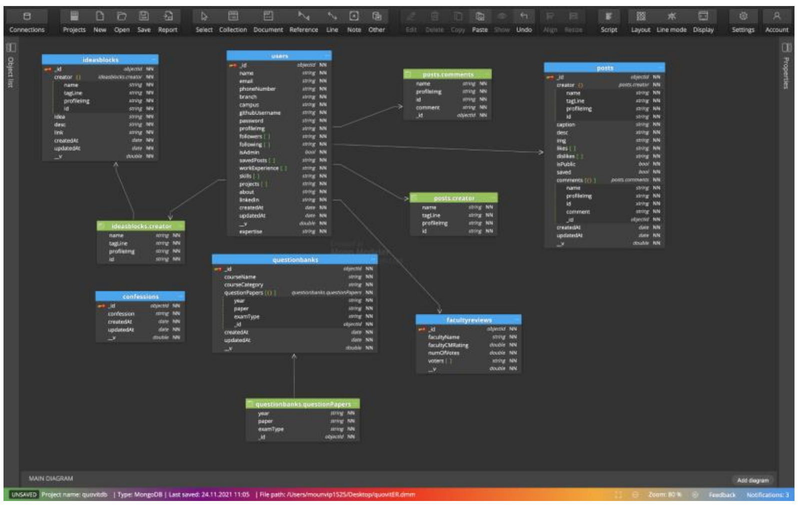
GET http://localhost:8000/posts/profile/619caf5ae8ca95b0fa19df08

This request does not have a body

Body — Status: 200 OK Time: 7.25 s Size: 794.42 KB

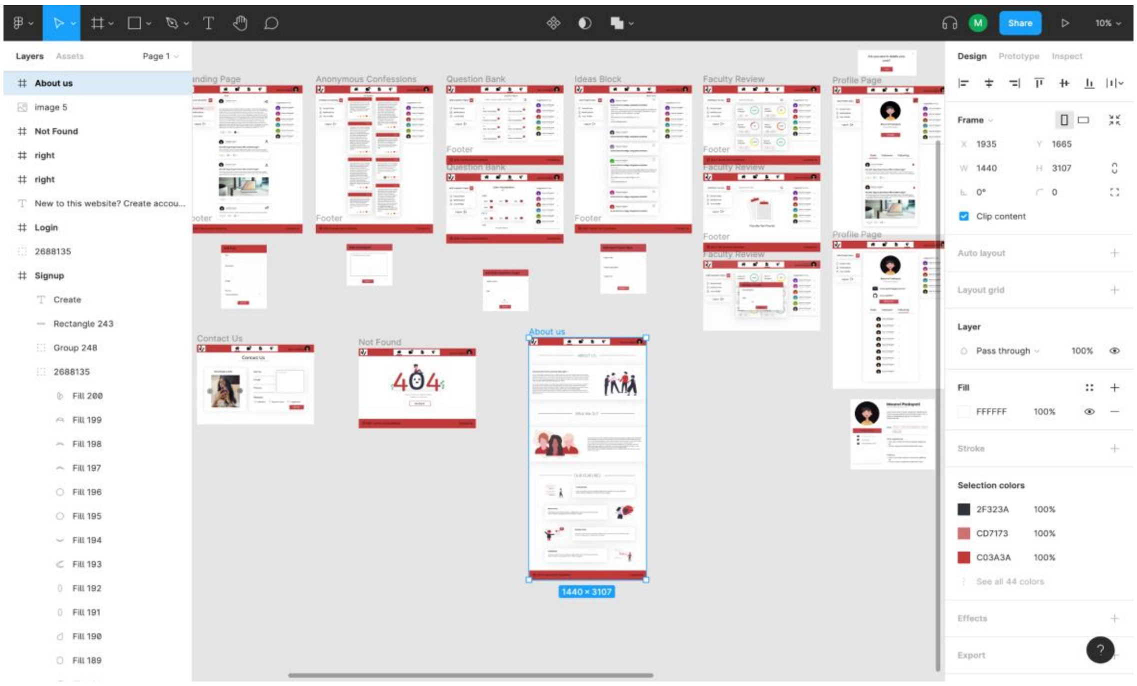
```
{
    "userPosts": [
        {
            "creator": {
                "name": "Mounvi Podapati",
                "tagLine": "CSE , VIT Vellore",
                "profileImg": "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAWMAAAGkCAMAAADT8bTfAAAABGdBTUEAALGPC/
                    xhBQAAAAFzUkdCAK7OHOkAAADnUExURQgICM/DshgYIdq/oQEBASwkJ+bg1AYGBgMDAwEBAde5mjAsMzUyOCgoMUA
                    +RhOeKMe6q807sxERGzk3PuPcz7uuoNnTyeLZyGVEL2g8TVBLT8GzpHRONsvEvdTIuZ1uTbqIZDMkJ66BZbKqpKd3U6qVg41jSQoKFZRnR7m
                    nlb6TdiIjLbGhlNXMwH9waMORaIJZPd3Twcusk2xkYtHMyLJ/V93b2Onl2s2efVcvPykQF4V7d7qdh4BcSkUpJ5iOh2RMQl5YWZiDdFo
                    +L1Q4Kp5zX3NcVUk+PKOak8A3N9C2nHNQRQwHDZ+Vt1AAAAAKdFJOUwL///8B//8MGC3oPGCOAAAgAElEQVR42oyaW2/
                    jOBKFW2MIPQBFRYGgFwkTQLPwg8eGHqw322i3gkW2x///
                    By3ryiIlJ03KujjBDPDl9OGpor59C6MPY4HxM59hfsTPFL6CMZkjfPgeL1MYeLLjCgeczJj7Jfxv
                    +nmua7j7atQw82M92haOGs7JqGS8rqocT0/o6jzdwRHuZcitg29LX5auPMKlhLv4+Z0x8mUsx28y+v/1/c9l
                    +UmclznOD54yJiU8Ed55izABvgLheL1O8JcMBzAm3J8jVsqMdwsx02XUhjBPF1i3iNYDUURLzBkxYoYrnwNVFyZcvWfG4XcA7xeEX1LKF4Qc
                    Gc+B8M8A9+MDlfmRqMIAuh9zSvD6DpIMp/d83093+MSxly/2Yf6630HHYQLjpU8Q1/
                    yJT6JkAsuEN6TcEuJEyFVLYFnKKGlfqY7DfZgoVx6e4NIZBIyoScJByzyY8tEcJfySX1EeScmX8TD
                    +PgWUIj8cyo0mnYBWHKf9rzBPeHPa38IRTjoPpzBvh0P4wBnGfyYS7/I7XoGcScMEuH
                    +qY9IwMq4YMenYAWmkDCcvDuFJyJUnJTt8CVCu4xuvOvbMzxvO2+OY6BgJwB0PYfz9
                    +zX8c1ewf78LUtAfynFv4cZxg09AG8bphmhvB5p6OR98dBPqmL3iKdjkC+MV/RM7RsxoFuLDz8h17EjVAByZkjVXRBh+Lh6BqIE/
                    KlwE6iNCeBoTxl5OLPWLYA4CZsbjj+/M+E9cnFC28E8e/43v3/dm/
                    A2nbPwDkrOZuvixfFnBh0D4TJQHZExm8Ztr3tKrC1vPEPEaQ87WvIoYk37JkEnIvNChNYuKmTJzdqxgMg1yOqtf
                    +PELQfWJmFHGiSMD5HH88ed3Zvw+TegM+3AEqIR3gy7A3d8S7TLagBGL8xzgBjwfuvOSO3fGK5YvRMxWkTpFnxiEMOf1rharSJOFzzRhkB
                    +jigmv8WOnGgbgJQtZ5KpazazCpZgvouQXRh5iRc4YLQHA3jcB3xDxjeieBOlBqTLhM9I9s4YD2q4743mgXLEkXlE/
                    ua5XvU2nkGBBsaLK4xuKmXCjAUueqAgp+oejoKGYwZZLXvS8LHpHs/
```

MongoDB Fields view:

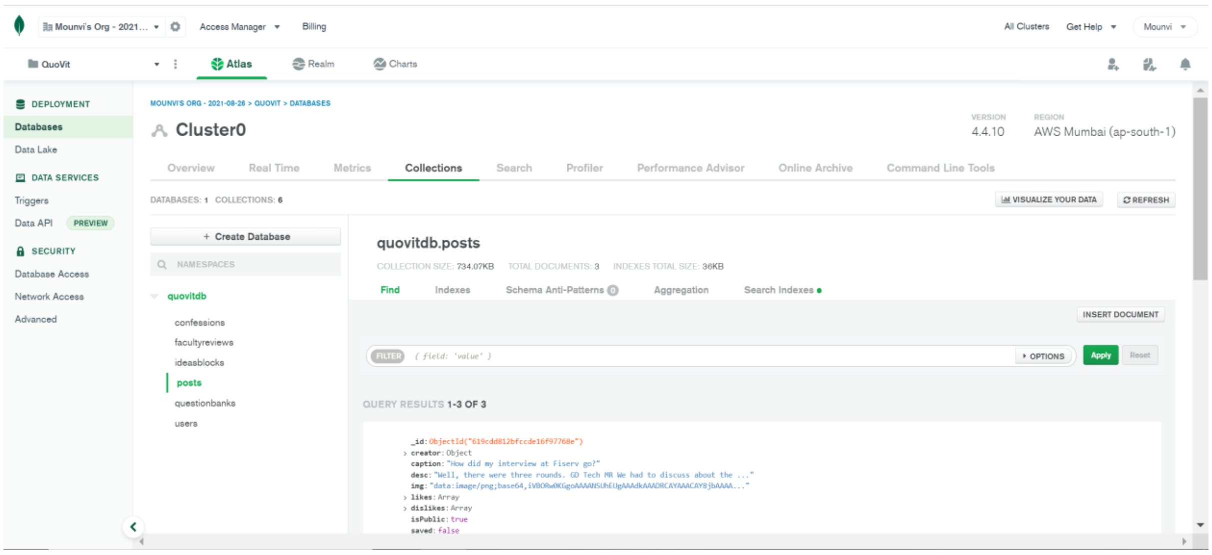


Project name: quovitdb | Type: MongoDB | Last saved: 24.11.2021 11:05 | File path: /Users/mounvip1525/Desktop/quovitER.dmm

Figma design: https://www.figma.com/file/RlwqrzDyETh2TUaAVy7xir/QuoVit?node-id=0%3A1



MongoDB Atlas:



```
{ field: 'value' }
```

```
  _id: ObjectId("619cdd812bfccde16f97768e")
> creator: Object
  caption: "How did my interview at Fiserv go?"
  desc: "Well, there were three rounds. GD Tech MR We had to discuss about the ..."
  img: "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAdkAAADRCAYAAACAY8jbAAAA..."
> likes: Array
> dislikes: Array
  isPublic: true
  saved: false
```

Video Demonstration link: https://youtu.be/zYL1du5k2Vs

FRONTEND CODE:

Landing.js

```
import React, { useEffect } from "react";
import { getPosts } from "../../Actions/posts";
import { useDispatch, useSelector } from "react-redux";
import MainLayout from "../../Components/Structure/Main";
import "./css/Landing.css";
import LandingCard from "../../Components/Cards/LandingCard";
import AddPostModal from "./AddPostModal";

export default function Landing(){
  const dispatch = useDispatch();
  const {posts,savedPosts} = useSelector((state) => state.posts);
  const auth = useSelector((state)=>state.auth)
  useEffect(() => {
    dispatch(getPosts(auth._id));
  }, [dispatch,auth]);

    return (
        <MainLayout type="landing">
          <AddPostModal />
          <div className="landing-main">
            {posts.map((post) => (
              <LandingCard
                post={post}
                saved={savedPosts.some(sPost=>sPost._id === post._id)}
              />
            ))}
          </div>
        </MainLayout>
    );
}
```

landingCard.js

```
import React, { useState } from "react";
import { useDispatch , useSelector } from "react-redux";
import { useHistory } from "react-router-dom"
import User from "../User/User.jsx";
import "./css/LandingCard.css";
import { addComment, deletePost, dislikePost, likePost, toggleSavePost } from "../../Actions/posts";
import DeleteIcon from '@material-ui/icons/Delete';
import ArrowDropUpIcon from '@mui/icons-material/ArrowDropUp';
import ArrowDropDownIcon from '@mui/icons-material/ArrowDropDown';
import CommentIcon from '@mui/icons-material/Comment';
import TurnedInNotIcon from '@mui/icons-material/TurnedInNot';
import { Send, TurnedIn } from "@material-ui/icons";
import Empty from "../Empty/Empty.js";
import LockIcon from '@mui/icons-material/Lock';
```

```
export default function LandingCard(props) {
  const history = useHistory()
  const [fullText,setFullText] = useState(true)
  const [showComments,setShowComments] = useState(false)
  const [comment,setComment] = useState("")
  const { creator , caption , desc , likes , dislikes , img , _id, comments, isPublic } = props.post;
  const [save,setSave] = useState(props.saved)
  const dispatch = useDispatch();
  const auth = useSelector((state)=>state.auth)
  const handleLikeClick = () => {
    if(auth._id){
      dispatch(likePost(_id,auth._id));
    } else {
      history.push("/Login")
    }
  };
  const handleDislikeClick = () => {
    if(auth._id){
      dispatch(dislikePost(_id,auth._id));
    } else {
      history.push("/Login")
    }
  }
  const handleSaveClick = () => {
    if(auth._id){
      setSave(!save)
      dispatch(toggleSavePost(_id,auth._id))
    } else {
      history.push("/Login")
    }
  }
  const handleDelete = () => {
      dispatch(deletePost(_id,auth._id))
  }
  const handleCommentClick = () => {
    setShowComments(!showComments)
  }
  const postComment = () => {
    if(!auth._id){
      history.push("/Login")
    } else if(!comment) {
      document.getElementById("comment").focus()
    } else {
      dispatch(addComment(_id,auth._id,comment))
    }
setComment("")
  }
  const likeCircleColors = ["#91C196","#C5D226", "#F6E015","#DA6767"];
  let diff = likes.length - dislikes.length
  const bgcolor = diff >= 10 ? 0 : diff>= 5 ? 1 : diff>= 3 ? 2 : 3;
```

```
 return (
   <div className="landingCard">
     {props.remove ? <div className="bin-post"><DeleteIcon onClick={handleDelete}/></div> :
     !isPublic ? <div className="bin-post bin-post2"><LockIcon /></div> : ""}
     <User user={creator} />
     <h3>{caption}</h3>
     <div>
       <div className="desc-div">
         {(desc.length>200 && fullText) ?
             <p style={{fontSize:"small"}}>{desc.substring(0,200)} &nbsp;<span
onClick={()=>{setFullText(false);console.log("hello")}}>More</span></p> :
             <p style={{fontSize:"small"}}>{desc}</p>
         }
       </div>
       {img &&
       <div className="post-img">
         <img src={img} alt="pic" />
       </div>}
       <div className="lcs-div">
         <div>
         <ArrowDropUpIcon onClick={handleLikeClick} />
         <div className="like-circle" style={{borderColor:likeCircleColors[bgcolor]}}>
           {diff}
         </div>
         <ArrowDropDownIcon onClick={handleDislikeClick} />
         </div>
         <div>
           <CommentIcon style={{marginRight:"0.3rem"}} onClick={handleCommentClick}/>
           {/* <p>{comments}</p> */}
           <p>{comments.length}</p>
         </div>
         <div>
           {save ?
           <TurnedIn style={{marginRight:"0.3rem"}} onClick={handleSaveClick} /> :
           <TurnedInNotIcon style={{marginRight:"0.3rem"}} onClick={handleSaveClick} />}
           <p>{save ? "Saved" : "Save"}</p>
         </div>
       </div>
     </div>
     {showComments &&
     <div className="comment-box">
       <div>
         <input type="text" name="comment" id="comment" className="edit-ip" placeholder="Add
your comment" value={comment} onChange={(e)=>setComment(e.target.value)}/>
         <button onClick={postComment}><Send /></button>
       </div>
       <div>
       {comments.length > 0 ? comments.map(c=> <User user={c} />) : <Empty msg="Be the first one to
comment" index={4} small={true}/>}
       </div>
     </div>}
```

```
    </div>
  );
}
```

API.js

```
import axios from 'axios'
const backend_url = 'http://localhost:8000/'

export const signUp = (user) => axios.post(`${backend_url}auth/signup`,user)
export const signIn = (user) => axios.post(`${backend_url}auth/signin`,user)

export const fetchPosts = (id) => axios.get(`${backend_url}posts/${id}`)
export const fetchSavedPosts = (id) => axios.get(`${backend_url}posts/saved/${id}`)
export const toggleSavePost = (postId,userId) =>
axios.put(`${backend_url}posts/saved/${userId}`,{postId})
export const likePost = (postId,userId) => axios.put(`${backend_url}posts/like/${postId}`,{userId})
export const dislikePost = (postId,userId) => axios.put(`${backend_url}posts/dislike/${postId}`,{userId})
export const addPost = (post,id) => axios.post(`${backend_url}posts/${id}`,post)
export const addComment = (postId,userId,comment) =>
axios.put(`${backend_url}posts/comment/${postId}/${userId}`,{comment})

export const profileDetails = (id) => axios.get(`${backend_url}posts/profile/${id}`)
export const editProfileDetails = (id,body) => axios.put(`${backend_url}posts/profile/${id}`,body)
export const deletePost = (postId,userId) => axios.delete(`${backend_url}posts/${postId}/${userId}`)

export const followUser = (userId,followId) =>
axios.put(`${backend_url}user/follow/${userId}`,{userId:followId})
export const unfollowUser = (userId,followId) =>
axios.put(`${backend_url}user/unfollow/${userId}`,{userId:followId})
export const getSuggestedUsers = (id) => axios.get(`${backend_url}user/suggestions/${id}`)

export const getMentors = () => axios.get(`${backend_url}user/getMentors`)

export const fetchConfessions = () => axios.get(`${backend_url}confessions`)
export const addConfession = (newConfession) =>
axios.post(`${backend_url}confessions/add`,{confession:newConfession})

export const fetchFacultyReviews = () => axios.get(`${backend_url}facultyReviews`)
export const addFaculty = (newFaculty,id) =>
axios.post(`${backend_url}facultyReviews/${id}`,newFaculty)
export const rateFaculty = (userId,facId,rating) =>
axios.put(`${backend_url}facultyReviews/rate/${userId}`,{facId,facultyRating:rating})

export const fetchCourses = () => axios.get(`${backend_url}questionBank`)
export const fetchPapersByCourse = (courseName) =>
axios.get(`${backend_url}questionBank/course/${courseName}`)
export const downloadPaper = (courseName,id) =>
axios.get(`${backend_url}questionBank/download/${courseName}/${id}`,{responseType: 'blob'})
export const uploadPaper = (courseName,courseCategory,examType,year,formData) =>
axios.post(`${backend_url}questionBank/${courseName}/${courseCategory}/${examType}/${year}`,for
mData)
```

```
export const fetchIdeas = () => axios.get(`${backend_url}ideasBlock/`)
export const addIdea = (idea,id) => axios.post(`${backend_url}ideasBlock/${id}`,idea)
```

Actions.js

```
import * as api from "../API";

export const getPosts = (id) => async (dispatch) => {
  dispatch({ type: "SET_LOADING" })
  try {
    if(id === null)
        id = "619ca53ce8ca95b0fa19defd"
    const { data } = await api.fetchPosts(id);
    dispatch({ type: "FETCH_ALL_POSTS", payload: data });
  } catch (error) {
    alert(error)
  }
};

export const getSavedPosts = (id) => async (dispatch) => {
  dispatch({ type: "SET_LOADING" })
  try {
    if(id === null)
        id = "619ca53ce8ca95b0fa19defd"
    const { data } = await api.fetchSavedPosts(id);
    dispatch({ type: "FETCH_SAVED_POSTS", payload: data });
  } catch (error) {
    alert(error.message);
  }
};

export const toggleSavePost = (postid,userid) => async (dispatch) => {
  try {
    if(userid === null)
      alert("login first")
    else {
  dispatch({ type: "SET_LOADING" })
    const { data } = await api.toggleSavePost(postid,userid);
    dispatch({ type: "TOGGLE_SAVE_POST", payload: data });
    }
  } catch (error) {
    alert(error);
  }
};

export const likePost = (postid,userid) => async (dispatch) => {
    try {
      if(userid === null)
```

```
        alert("login first")
        else {
  dispatch({ type: "SET_LOADING" })
      const { data } = await api.likePost(postid,userid);
      dispatch({ type: "LIKE_POST", payload: data });
        }
    } catch (error) {
      alert(error.message);
    }
  };


  export const dislikePost = (postid,userid) => async (dispatch) => {
    try {
      if(userid === null)
        alert("login first")
        else {
  dispatch({ type: "SET_LOADING" })
      const { data } = await api.dislikePost(postid,userid);
      dispatch({ type: "DISLIKE_POST", payload: data });
        }
    } catch (error) {
      alert(error.message);
    }
  };

export const addPost = (post,id) => async (dispatch) => {
  try {
      if(id){
  dispatch({ type: "SET_LOADING" })
        const { data } = await api.addPost(post,id);
        console.log(data)
        dispatch({type: "ADD_POST", payload: data});
      }
  } catch(error) {
    alert(error);
  }
}

export const profileDetails = (id) => async (dispatch) => {
  try {
    if(id){
      dispatch({ type: "SET_LOADING" })
      const { data } = await api.profileDetails(id);
      dispatch({type:"FETCH_PROFILE_DETAILS", payload:data})
    } else {
      alert("login first")
    }
  } catch (err) {
    alert(err)
  }
```

```
}
export const updateProfileDetails = (id,userId,edits) => async (dispatch) => {
  dispatch({ type: "SET_LOADING" })
  try {
    if(id !== userId){
      alert("You can only edit your own post")
    } else {
      const p = edits.proj.split(",");
      const s = edits.skil.split(",");
      const w = edits.workExp.split(",")
      const { data } = await api.editProfileDetails(id,{projects:p,skills:s,workExperience:w});
      dispatch({type:"FETCH_PROFILE_DETAILS",payload:data})
    }
  } catch(err){
    alert(err)
  }
}

export const deletePost = (postid,userId) => async (dispatch) => {
  try {
    if(!userId){
      alert("Login first")
    } else {
  dispatch({ type: "SET_LOADING" })
      const { data } = await api.deletePost(postid,userId);
      dispatch({type:"DELETE_POST",payload:data})
    }
  } catch(err){
    alert(err)
  }
}

export const addComment = (postId,userId,comment) => async (dispatch) => {
  try {
    if(!userId){
      alert("Login first")
    } else {
  dispatch({ type: "SET_LOADING" })
      const { data } = await api.addComment(postId,userId,comment);
      dispatch({ type:"ADD_COMMENT" , payload:data })
    }
  } catch(err){
    alert(err)
  }
}
```

Confessions.js

```
import React, { useEffect } from "react";
import ConfessionCard from "../../Components/Cards/ConfessionCard";
import MainLayout from "../../Components/Structure/Main";
import "./css/Confessions.css";
import { getConfessions } from "../../Actions/confessions";
import { useDispatch, useSelector } from "react-redux";
import AddConfession from "./AddConfession";

export default function Confessions() {
  const dispatch = useDispatch();
  const { confessions } = useSelector((state) => state.confessions);
  useEffect(() => {
    dispatch(getConfessions());
  }, [dispatch]);
  return (
    <MainLayout type="confessions">
      <AddConfession />
      <div className="confessions-cont">
        <div className="c-1">
          {confessions.map((confession, index) =>
            index % 2 === 0 ? (
              <ConfessionCard
                date={confession.date}
                confession={confession.confession}
                id={confession.id}
                index={index}
              />
            ) : (
              ""
            )
          )}
        </div>
        <div className="c-2">
          {confessions.map((confession, index) =>
            index % 2 !== 0 ? (
              <ConfessionCard
                date={confession.createdAt}
                confession={confession.confession}
                id={confession.id}
                index={index}
              />
            ) : (
              ""
            )
          )}
        </div>
      </div>
    </MainLayout>
  );
}
```

FacultyReviews.js

```
import React, {useEffect,useState} from "react";
import MainLayout from "../../Components/Structure/Main";
import FacultyReviewCard from "../../Components/Cards/FacultyReviewCard";
import AddFacultyModal from "./AddFacultyModal";
import { getFacultyReviews } from "../../Actions/facultyReviews";
import { useDispatch, useSelector } from "react-redux";
import RateFacultyModal from "./RateFacultyModal";
import Search from "./Search";
import "./css/FacultyReview.css";

export default function FacultyReview() {
  const [rateShow,setRateShow] = useState(false);
  const [activeFaculty,setAcitveFaculty] = useState({id:"",name:""})
  const dispatch = useDispatch();
  const {facultyReviews} = useSelector((state) => state.facultyReviews);
  useEffect(() => {
    dispatch(getFacultyReviews());
  }, [dispatch]);
  const handleRateShow = (id,name) => {
    setAcitveFaculty({id,name});
    setRateShow(true)
  }
  return (
    <MainLayout type="facultyReviews">
      <Search faculties={facultyReviews} />
      <AddFacultyModal />
      <RateFacultyModal showModal={rateShow} activeFaculty={activeFaculty}
closeModal={()=>setRateShow(false)}/>
      <div className="fac-main">
        <div className="fac-cards">
          {facultyReviews.map((faculty) => (
            <FacultyReviewCard faculty={faculty} handleShow={handleRateShow}/>
          ))}
        </div>
      </div>
    </MainLayout>
  );
}
```

IdeasBlock.js

```
import React, { useEffect } from "react";
import { getIdeas } from "../../Actions/ideasBlock";
import { useDispatch, useSelector } from "react-redux";
import MainLayout from "../../Components/Structure/Main";
import IdeaCard from "../../Components/Cards/IdeaCard";
import "./css/IdeasBlock.css";
import AddIdeaModal from "./AddIdeaModal";

export default function IdeasBlock() {
  const dispatch = useDispatch();
  const { ideas } = useSelector((state) => state.ideas);
  useEffect(() => {
    dispatch(getIdeas());
  }, [dispatch]);
  return (
    <MainLayout type="ideasBlock">
      <AddIdeaModal />
      <div className="ideas-main">
        {ideas.map((idea) => (
          <IdeaCard idea={idea} />
        ))}
      </div>
    </MainLayout>
  );
}
```

questionBank.js

```
import React , { useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";
import CourseCard from "../../Components/Cards/CourseCard";
import MainLayout from "../../Components/Structure/Main";
import AddPaperModal from "./AddPaperModal";
import { getCourses } from "../../Actions/questionBank";
import Search from "./Search";
import "./css/QuestionBank.css";

export default function QuestionBank() {
  const dispatch = useDispatch();
  const { courses } = useSelector((state) => state.questionBank);
  useEffect(() => {
    dispatch(getCourses());
  }, [dispatch]);
  return (
    <MainLayout type="questionBank">
      <AddPaperModal />
      <Search courses={courses} />
      <div className="course-container">
        {courses.map((course) => (
          <CourseCard course={course} />
        ))}
      </div>
    </MainLayout>
  );
}
```

BACKEND CODE

Posts.js

```
import users from "../models/user.js";
import posts from "../models/posts.js";

const get_user_posts = async (userId) => {
   try {
      let userPosts = [], uniquePostIds = [], uniquePosts = [], savedPosts = [];
      if (userId) {
         const currentUser = await users.findById(userId);
         userPosts = await posts.find({ "creator.id": currentUser._id });
         const friendPosts = await Promise.all(
            currentUser.following.map((friendId) => {
               return posts.find({ "creator.id": friendId });
            })
         );
         friendPosts.map(fPost => {
            fPost.length > 0 ? fPost.map(f => userPosts.push(f)) : null
         })
         savedPosts = await Promise.all(
            currentUser.savedPosts.map(post => {
               return posts.findById(post);
            })
         );
      }
      const publicPosts = await posts.find({ isPublic: true })
      userPosts.push(...publicPosts)
      userPosts.map(post => {
         if (!uniquePostIds.includes(post._id.toString())) {
            uniquePostIds.push(post._id.toString())
            uniquePosts.push(post)
         }
      })
      return {uniquePosts,savedPosts}
   } catch (err) {
      console.log(err)
   }
}
//create a post
const create_post = async (req, res) => {
   let uTag;
   const userId = req.params.id;
   users.findById(userId).then(result => {
      try {
         uTag = `${result.branch} , VIT ${result.campus}`
         posts.create({ ...req.body, creator: { name: result.name, tagLine: uTag, profileImg:
result.profileImg, id: result._id } }, (err, data) => {
            if (err) {
               res.status(500).send(err);
```

```
                } else {
                    res.status(201).send(data);
                }
            });
        } catch (err) {
            console.log(err)
            res.status(500).send(err);
        }
    })
};

//Get timeline posts
const get_timeline_posts = async (req, res) => {
    try {
        const uniquePosts = await get_user_posts(req.params.id)
        res.send(uniquePosts)
    } catch (err) {
        res.send(err)
    }
};

//delete a post
const delete_post = async (req, res) => {
    try {
        const post = await posts.findById(req.params.postId);
        if (post.creator.id === req.params.userId) {
            await post.deleteOne();
            const pd = await fetch_user_profile(req.params.userId)
            res.status(200).send(pd)
        } else {
            res.status(403).send("you can delete only your post");
        }
    } catch (err) {
        console.log(err)
        res.status(500).send(err);
    }
};

//like a post
const like_post = async (req, res) => {
    try {
        const post = await posts.findById(req.params.id);
        if (!post.likes.includes(req.body.userId)) {
            await post.updateOne({ $push: { likes: req.body.userId } });
        }
        if (post.dislikes.includes(req.body.userId)) {
            await post.updateOne({ $pull: { dislikes: req.body.userId } });
        }
        const uniquePosts = await get_user_posts(req.body.userId)
        res.send(uniquePosts)
    } catch (err) {
        console.log(err)
        res.status(500).send(err);
    }
};

//Dislike a post
const dislike_post = async (req, res) => {
    try {
        const post = await posts.findById(req.params.id);
        if (!post.dislikes.includes(req.body.userId)) {
            await post.updateOne({ $push: { dislikes: req.body.userId } });
        }
        if (post.likes.includes(req.body.userId)) {
            await post.updateOne({ $pull: { likes: req.body.userId } });
        }
        const uniquePosts = await get_user_posts(req.body.userId)
        res.send(uniquePosts)
    } catch (err) {
        res.status(500).send(err);
    }
};

//update a post
const update_post = async (req, res) => {
    try {
        const post = await posts.findById(req.params.id);
        if (post.creator.id === req.body.userId) {
            await post.updateOne({ $set: req.body });
            res.status(200).send("the post has been updated");
        } else {
            res.status(403).send("you can update only your post");
        }
    } catch (err) {
        console.log(err)
        res.status(500).send(err);
    }
};

//get a post
const get_post = async (req, res) => {
    try {
        const post = await posts.findById(req.params.id);
        res.status(200).send(post);
    } catch (err) {
        res.status(500).send(err);
    }
};

//save posts
const toggle_save_post = async (req,res) => {
    try {
        const user = await users.findById(req.params.id)
        !user.savedPosts.includes(req.body.postId) ?
            await user.updateOne({ $push: { savedPosts : req.body.postId }}) :
            await user.updateOne({ $pull: { savedPosts : req.body.postId }})
            const posts = await get_user_posts(req.params.id)
            res.send(posts)
    } catch(err){
        console.log(err)
        res.status(500).send(err)
    }
}
//get saved posts
const get_saved_posts = async (req,res) => {
    try {
        const user = await users.findById(req.params.id)
        const savedPosts = await Promise.all(
            user.savedPosts.map(post => {
                return posts.findById(post);
            })
        );
        res.status(200).send(savedPosts)
    } catch(err){
        res.status(500).send(err)
    }
}

const fetch_user_profile = async (id) => {
    try{
        const currentUser = await users.findById(id);
        const userPosts = await posts.find({ "creator.id": currentUser._id });
        const following = await Promise.all(
            currentUser.following.map(async (friendId) => {
                let user = await users.findById({_id:friendId});
                let tagline = `${user.branch} , VIT ${user.campus}`
                let { name , profileImg ,_id} = user;
                return {name,profileImg,tagline,_id}
            }))
        const followers = await Promise.all(
            currentUser.followers.map(async (friendId) => {
                let user = await users.findById({_id:friendId});
                let tagline = `${user.branch} , VIT ${user.campus}`
                let { name , profileImg ,_id} = user;
                return {name,profileImg,tagline,_id}
            }))
        return {userPosts,currentUser,followers,following}
    } catch(err){
        console.log(err)
    }
}

const get_profile_details = async (req,res) => {
    try{
        const pd = await fetch_user_profile(req.params.id)
        res.status(200).send(pd)
    } catch(err){
        res.status(500).send(err)
    }
}

const update_profile_details = async (req, res) => {
    try {
        const user = await users.findById(req.params.id);
        await user.updateOne({ $set: req.body });
        const pd = await fetch_user_profile(req.params.id)
        res.status(200).send(pd);
    } catch (err) {
        res.status(500).send(err);
    }
};

const add_comment = async (req,res) => {
    try {
        const post = await posts.findById(req.params.postId)
        const user = await users.findById(req.params.userId)
        await post.updateOne({ $push: { comments: { name : user.name, profileImg : user.profileImg, id:
user.id, comment: req.body.comment } }});
        const uniquePosts = await get_user_posts(req.params.userId)
        res.send(uniquePosts)
    } catch(err) {
        console.log(err)
        res.status(500).send(err)
    }
}
export { create_post, update_post, delete_post, get_post, get_timeline_posts, like_post, dislike_post,
get_saved_posts , toggle_save_post , get_profile_details, update_profile_details, fetch_user_profile,
add_comment }
```

postModel.js

```
import mongoose from "mongoose";

const postSchema = new mongoose.Schema(
  {
    creator:{
      name:String,
      tagLine: String,
      profileImg:String,
      id:String
    },
    caption: {
      type: String,
      required: true,
    },
    desc: {
      type: String
    },
    img: {
      type: String
    },
    likes: {
      type: Array,
      default: [],
    },
    dislikes: {
      type: Array,
      default: [],
    },
    isPublic:{
      type:Boolean,
      default:true
    },
    saved:{
      type:Boolean,
      default:false
    },
    comments: [
      {
        name:String,
        profileImg:String,
        id:String,
        comment: String
    }
  ],
  },
  { timestamps: true }
);

export default mongoose.model("post", postSchema);
```

postRoutes.js

```
import express from "express";
import { create_post , update_post , delete_post , get_post , get_timeline_posts , like_post ,
dislike_post, get_saved_posts, toggle_save_post, get_profile_details, update_profile_details,
add_comment  } from "../controllers/postController.js";

const router = express.Router();

//create a post
router.post("/:id", create_post);
//update a post
router.put("/:id", update_post);
```

```
//delete a post
router.delete("/:postId/:userId", delete_post);
//like a post
router.put("/like/:id", like_post);
// dislike a post
router.put("/dislike/:id", dislike_post);
//get a post
// router.get("/:id", get_post);
//get timeline posts
router.get("/:id", get_timeline_posts);
//get saved posts
router.get("/saved/:id",get_saved_posts);
//toggle save posts
router.put("/saved/:id",toggle_save_post)
// user profile details
router.get("/profile/:id",get_profile_details);
//update profile details
router.put("/profile/:id",update_profile_details);
//add comment
router.put("/comment/:postId/:userId",add_comment)

export default router;
```

<u>users.js</u>

```
import users from "../models/user.js";
import { fetch_user_profile } from "./postController.js";
import bcrypt from "bcrypt";

//follow a user
const follow_user = async (req, res) => {
  if (req.body.userId !== req.params.id) {
    try {
      const user = await users.findById(req.params.id);
      const currentUser = await users.findById(req.body.userId);
      if (!user.following.includes(req.body.userId)) {
        await user.updateOne({ $push: { following: req.body.userId } });
        await currentUser.updateOne({ $push: { followers: req.params.id } });
        res.status(200).send("user has been followed");
      } else {
        res.status(403).send("you already follow this user");
      }
    } catch (err) {
      res.status(500).send(err);
      console.log(err);
    }
  } else {
    res.status(403).json("you cant follow yourself");
  }
};
```

```
//unfollow a user
const unfollow_user = async (req, res) => {
  if (req.body.userId !== req.params.id) {
    try {
      const user = await users.findById(req.params.id);
      const currentUser = await users.findById(req.body.userId);
      if (user.following.includes(req.body.userId)) {
        await user.updateOne({ $pull: { following: req.body.userId } });
        await currentUser.updateOne({ $pull: { followers: req.params.id } });
        const pd = await fetch_user_profile(req.params.id)
        res.status(200).send(pd)
        // res.status(200).send("user has been unfollowed");
      } else {
        res.status(403).send("you dont follow this user");
      }
    } catch (err) {
      res.status(500).send(err);
    }
  } else {
    res.status(403).send("you cant unfollow yourself");
  }
};

//get a user
const get_user = async (req, res) => {
  try {
    const user = await users.findById(req.params.id);
    const { password, updatedAt, createdAt, isAdmin, ...other } = user._doc;
    res.status(200).send(other);
  } catch (err) {
    res.status(500).send(err);
  }
};

//update user
const update_user = async (req, res) => {
  console.log(req.body.userId, req.params.id);
  if (req.body.userId === req.params.id || req.body.isAdmin) {
    if (req.body.password) {
      try {
        const salt = await bcrypt.genSalt(10);
        req.body.password = await bcrypt.hash(req.body.password, salt);
      } catch (err) {
        return res.status(500).send(err);
      }
    }
    try {
      const user = await users.findByIdAndUpdate(req.params.id, {
        $set: req.body,
      });
      res.status(200).json("Account has been updated");
    } catch (err) {
      console.log(err);
      return res.status(500).send(err);
    }
  } else {
    return res.status(403).send("You can update only your account!");
  }
};

//delete user
const delete_user = async (req, res) => {
  if (req.body.userId === req.params.id || req.body.isAdmin) {
    try {
      await users.findByIdAndDelete(req.params.id);
      res.status(200).send("Account has been deleted");
    } catch (err) {
      return res.status(500).send(err);
    }
  } else {
    return res.status(403).send("You can delete only your account!");
  }
};

//get all users
const users_index = (req, res) => {
  users
    .find()
    .then((result) => {
      res.status(200).send(result);
    })
    .catch((err) => {
      res.status(500).send(err);
    });
};

// get suggested users
const suggested_users_index = async (req, res) => {
  try {
    const user = await users.findById(req.params.id);
    const following = await Promise.all(
      user.following.map(async (friendId) => {
        let user = await users.findById({ _id: friendId });
        return user._id.toString();
      }))
    let all_users = await users.find()
    let suggestions = all_users.filter(user=>!following.includes(user._id.toString()));
    let final = suggestions.filter(s=>s._id.toString() !== req.params.id)
    final = final.map((user) => {
          let tagline = `${user.branch} , VIT ${user.campus}`
          let { name , profileImg ,_id } = user;
          return {name,profileImg,tagline,_id}
      })
    res.status(200).send(final)
  } catch(err){
    console.log(err)
    res.status(500).send(err)
  }
}
const get_mentors = async (req,res) => {
  try {
    let webDev = [] , pythonDev = [] , ml = [] , bct = [] , ds = [];
    const all_users = await users.find()
    all_users.map(user=>{
      user.expertise === "Web Development" ?
webDev.push({name:user.name,profileImg:user.profileImg,_id:user._id}) :
      user.expertise === "Python Developer" ?
pythonDev.push({name:user.name,profileImg:user.profileImg,_id:user._id}) :
      user.expertise === "Machine Learning" ?
ml.push({name:user.name,profileImg:user.profileImg,_id:user._id}) :
      user.expertise === "Blockchain Dev" ?
bct.push({name:user.name,profileImg:user.profileImg,_id:user._id}) :
      user.expertise === "Data Science" ?
ds.push({name:user.name,profileImg:user.profileImg,_id:user._id}) :
null
    })
    res.status(200).send({webDev,pythonDev,ml,bct,ds})
  } catch(err){
    console.log(err)
    res.status(500).send(err)
  }
}
export { update_user, delete_user, get_user, follow_user, unfollow_user, users_index,
suggested_users_index, get_mentors };
```

usersModel.js

```
import mongoose from "mongoose"

const UserSchema = new mongoose.Schema(
  {
    name: {
      type: String,
      required: true,
      min: 3,
      max: 50,
    },
    email: {
      type: String,
      required: true,
      max: 50,
      unique: true,
    },
    phoneNumber:{
        type: String,
        required: true,
        unique: true,
    },
    branch:{
        type: String,
        required: true
    },
    campus:{
        type: String,
        default: "Vellore"
    },
    githubUsername:String,
    password: {
      type: String,
      required: true,
      min: 8,
    },
    profileImg: {
      type: String,
      default: "",
    },
    followers: {
      type: Array,
      default: [],
    },
    following: {
      type: Array,
      default: [],
    },
    isAdmin: {
      type: Boolean,
      default: false,
    },
    savedPosts:{
        type:Array,
        default:[]
    },
    workExperience:{
      type:Array,
      default:[]
    },
    skills:{
      type:Array,
      default:[]
    },
    projects:{
      type:Array,
      default:[]
    },
        about:{
          type:String,
          default:""
        },
        linkedIn:{
          type:String,
          default:""
        },
        expertise: {
          type:String,
          default:null
        }
      },
      { timestamps: true }
    );

    export default mongoose.model("User", UserSchema);
```

userRoutes.js

```
import express from "express";
import {update_user,delete_user,get_user,follow_user,unfollow_user,users_index,
suggested_users_index, get_mentors} from "../controllers/usersController.js"

const router = express.Router()

//get all users
router.get("/",users_index)

router.get("/getMentors",get_mentors)

//update user
router.put("/:id", update_user);

//delete user
router.delete("/:id", delete_user);

//get a user
router.get("/:id", get_user);

//follow a user
router.put("/follow/:id", follow_user);

//unfollow a user
router.put("/unfollow/:id", unfollow_user);

//get suggested users
router.get("/suggestions/:id",suggested_users_index);

export default router;
```