

**A Report on**

## Warf Engine

Submitted for partial fulfillment of award of

**BACHELOR OF TECHNOLOGY**

degree

In

Computer Science & Engineering

By

ANEESH KUMAR(1900300130010)

ANMOL AHUJA (1900300130013)

AYUSH BHANDARI (1900300100056)

TARUN BANSAL (1900300100231)

MENTOR

**Prof. Shweta Chaku**



**INDERPRASTHA ENGINEERING COLLEGE, GHAZIABAD,**

**Dr. A P J ABDUL KALAM TECHNICAL UNIVERSITY LUCKNOW**

**2022-23**

**Certificate**

Certified that **Aneesh Kumar, Anmol Ahuja, Ayush Bhandari, Tarun Bansal** has carried out the project work presented in this report entitled **“Warf Engine”** for the award of **Bachelor of Technology** from Inderprastha Engineering College, Ghaziabad, under my supervision. The report embodies results of original work and studies carried out by Student himself/herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.

Prof. Shweta Chaku

Assistant Professor

Inderprastha Engineering College

Date: 21 December, 2022

ii

**Acknowledgement**

We take this opportunity to thank our teachers and friends who helped us throughout the project.

First and foremost I would like to thank my guide for the project (**Mr. Shweta Chaku, Assistant Professor, Computer Science & Engineering**) for his valuable advice and time during development of the project.

We would also like to thank **Dr. Vijai Singh (HoD, Computer Science Department**) for his constant support during the development of the project.

Aneesh Kumar Anmol Ahuja

1900300130010 1900300130013

Ayush Bhandari Tarun Bansal

1900300100056 1900300100231

iii

**Declaration**

*I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.*

*Signature:*

*Name:* Aneesh Kumar, Anmol Ahuja, Ayush Bhandari, Tarun Bansal

*Roll No.: 1900300130010, 1900300130013, 1900300100056, 1900300100231*

*Date: 23/12/22*

*iv*

# Abstract

WarfEngine is an open source platform for developers to create enterprise level web application in MERN stack. This platform already contains components like user management, roles management, module management, access management, content management with production ready setup which helps developers to develop their application in minimum cost.

v Table Of Contents

**CHAPTER NO. TITLE PAGE NO.**

[**ABSTRACT**](#_heading=h.gjdgxs) **5**

1. **INTRODUCTION Page No**
   1. Problem Definition 8
   2. Background about the project idea 8
   3. Objectives of proposed system 9
   4. Feasibility Study, need and significance 9
   5. Novelty of Project 10
   6. Technical Specification 10

1.6.1 Hardware and Software required

1. **LITERATURE REVIEW**
   1. Existing System - 11

Theoretical and methodological contributions to a particular topic of scholarly papers/ articles /books

1. **PROPOSED SYSTEM** 12

1. **SOFTWARE REQUIREMENT ANALYSIS** 
   1. Functional Requirements 13
      1. Use Case diagram 14
   2. Major Modules and their functionalities 15
2. **SYSTEM ANALYSIS & DESIGN**
   1. Sequence diagrams 16
   2. DFDs of the project 17
   3. Gantt Chart & Pert Chart (Wherever applicable) 17

6

1. **IMPLEMENTATION/CORE MODULE**
   1. Tables – explaining all fields and their data types for data base 18 used in project.
   2. Used Algorithms/Approaches for projects. 19
2. **RESULTS / OUTPUT** 20-21
3. **REFERENCES** 22
4. **CONCLUSIONS / RECOMMENDATIONS** 23
5. **APPENDICES**

10.1 Steps to execute/run/implement the project 24

10.2Coding if any Introduction 24-29

7

### Introduction

### 

### 1.1 Problem Definition

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 Background about the Project Idea

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 be able to 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 us to build something like this

8

### 1.3 Objective

To solve the problem statement , we 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 of 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 .

### Feasibility Study, need and Significance

* + - **Feasibility Study :** The primary specification of the project includes the basic knowledge of MERN stack, which in turn includes MongoDb + Express js + React js + Node js.
    - **Need :** The Project makes it convenient for the students to share their work on such platform and get reviews on the same.
    - **Significance** : It bridges the communication gap between the professors and students such that professors can provide their help to students and motivates them timely.

9

### Novelty of project

Our website will not only portray student’s projects but also serve as the learning platform for students where both professors and students will share their ideas as well as research papers and these research papers and ideas may serve as a foundation for new projects which will provide students an opportunity to learn and grow.

### Technical Specifications

* + 1. **Hardware**
       - Processor: Intel i5 or higher
       - Space On HDD: 5GB
       - RAM: 4GB
    2. **Software**
* Operating System: Windows 10
* Coding Language: Javascript
* IDE: VS Code

10

**2. Literature Review**

For our project work we have reviewed the following research papers :

[1] Hau Tran, “Developing a social platform based on MERN stack” The main goal of this research paper was to study the basic components of the highly popular MERN stack, 2021.

[2]Dominik Ertl TU Wien, Harald Krapfenbauer, “A case study of developing an IDE for embedded software using open source,”Proceedings of the Software Engineering Advances, 2009. ICSEA '09. Fourth International Conference, 2009.

[3] Rohit Tamrakar, [Niraj Wani](https://www.researchgate.net/profile/Niraj-Wani), “Design and Development of CHATBOT: A Review,” This paper reviews the technique, terminology, and different platforms used to design and develop the CHATBOT.

[4] Dominik Ertl TU Wien, Harald Krapfenbauer, “ A Case Study of Developing an IDE for Embedded Software Using Open Source,” This paper recommends developing a customized IDE based on open source software. We present a case study of developing such an IDE for the languages C#, Python and JavaScript.

[5]Mohammed Farik, Nilesh A. Lal, Shailendra Prasad,“Review Of Authentication Methods,” This Paper describes the importance and types of Authentication mechanism and how to implement those authentication mechanism to enhance security and reduce vulnerabilities.

[6]Jill Freyne,Shlomo Berkovsky , Elizabeth M. daly, Werner Geyer, “Social networking feeds:Recommending items of interest,” This Research Paper gives the detailed description about the working of feeds for social media platforms . This research paper shows how to predict best feed item relevance by combining short term interest models, exploiting previous viewing behavior of users, and long-term models, exploiting previous viewing of network actions.

11

**3.Proposed System**

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 .

###### 

###### 12

**4. Software Requirement Analysis**

**4.1 Functional Requirements**

The functional requirement of the project includes :

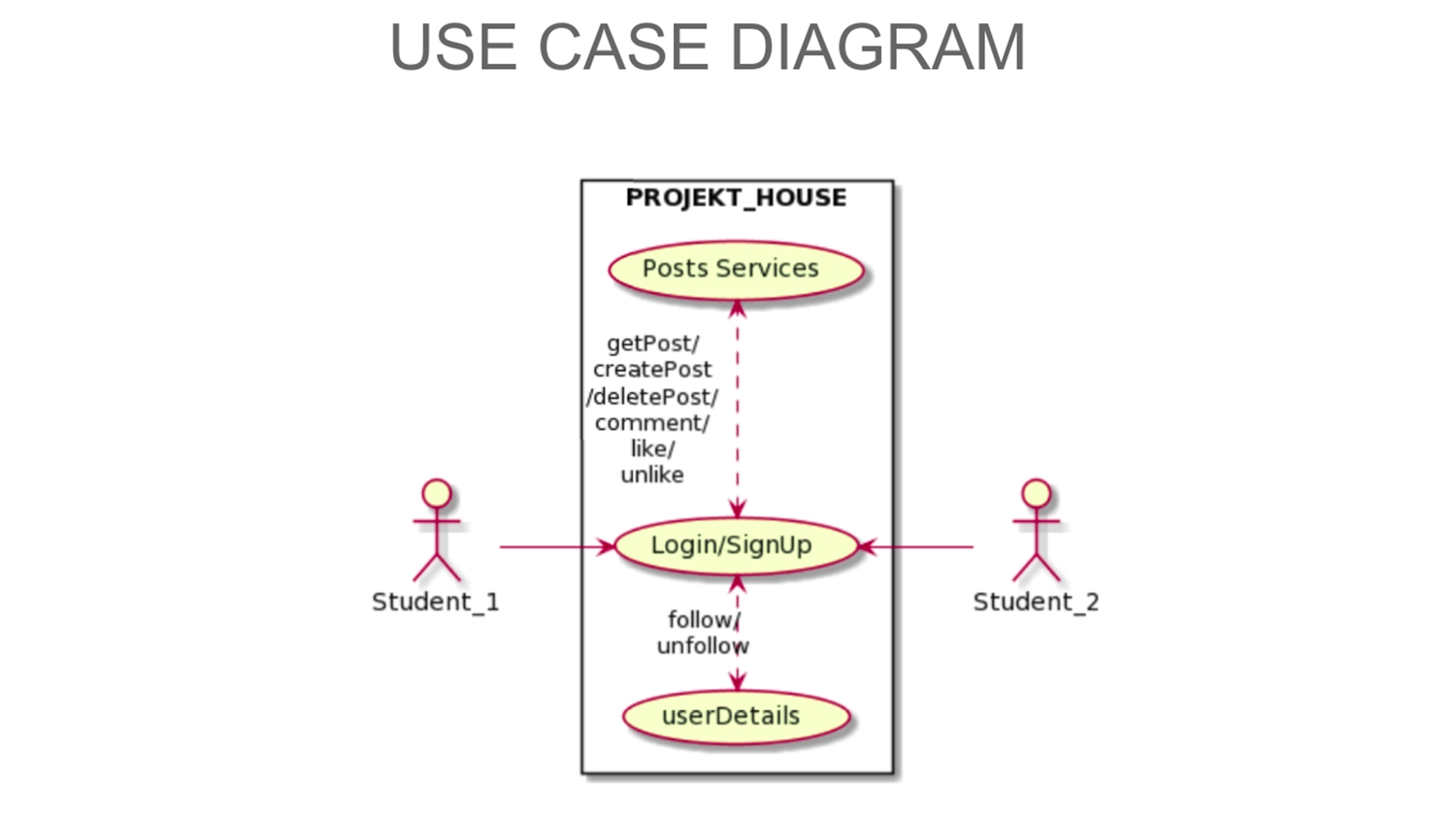
* Git : The git is installed for pushing the code and it also helps to control the version .
* Node Js : The project is entirely based on NodeJs . Hence,installing node is one of the key steps to run the project.
* Packages And Libraries

* MongoDB Compass and MongoDB Community Server :

These are required to support and run database server.

13

1. **Use Case Diagram**

****

14

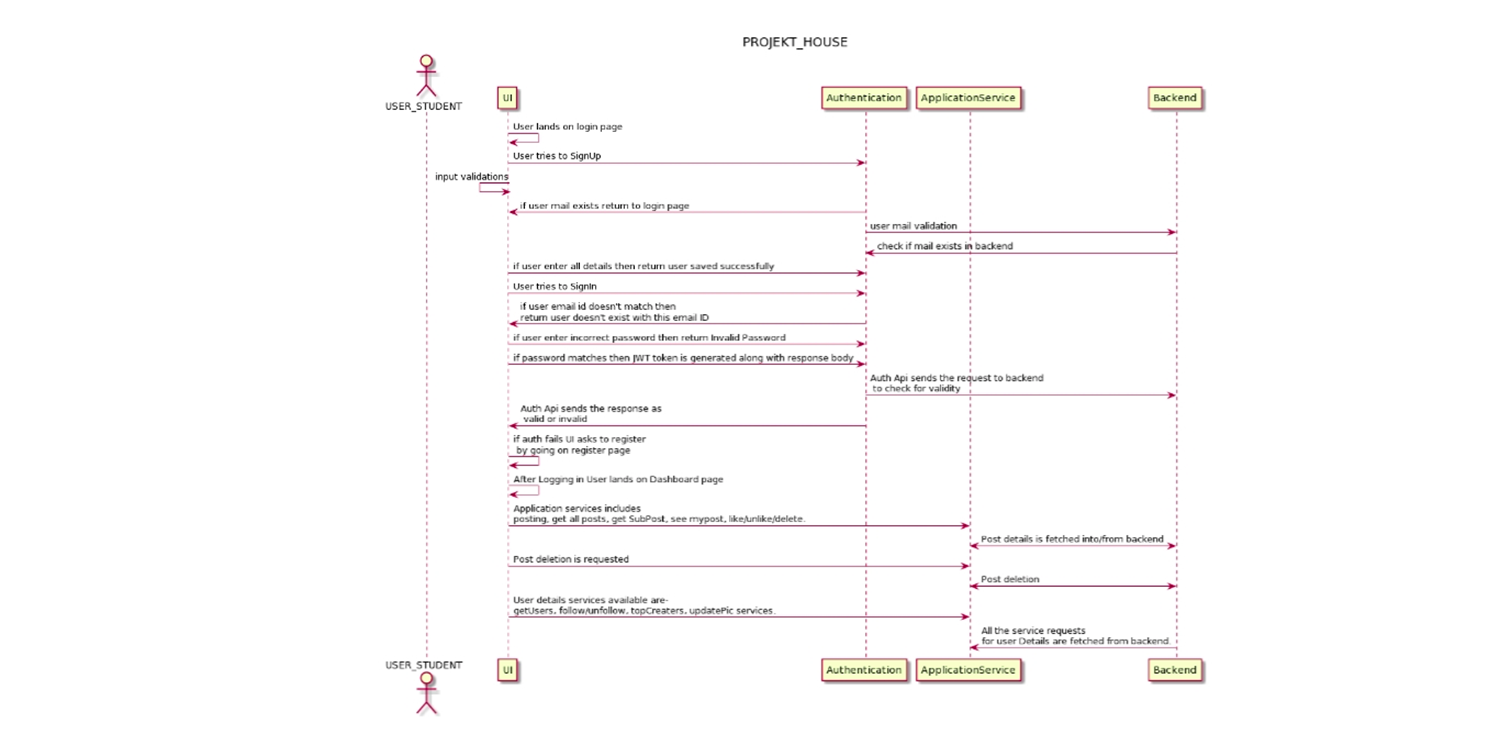
**4.2 Major Modules and their functionalities**

* **Authentication :** The project asks user id and password that are created during the registration as verification details for authentication of users.
* **Social Media Features:** The project provides most of the the functionalities that are provided by any social media website such as :
* creation of user profile after registering.
* option for creating and deleting posts.
* users can upload images of their project with description.
* follow / unfollow other users on the platform.
* feature to like , comment on any post.
* home feed showing posts from people that the user is following.
* **File Upload:** The project provides a feature to upload files like pdf and have different fields such as project/file name and description of uploaded file.
* **File List**: A list is visible to the users showing the title and description of the file and users also have the option to download them.
* **User search** : The project provides user the facility to search other users on the platform either by their username or email address.

15

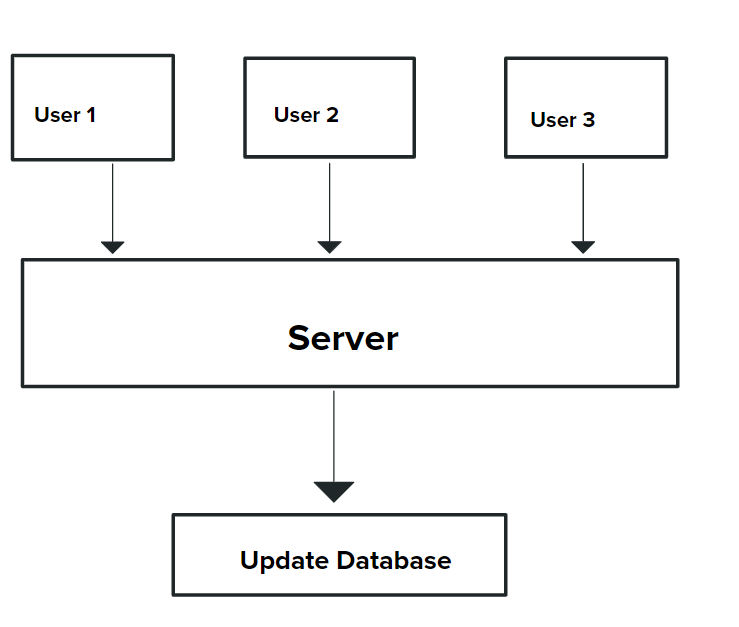
**5. System Analysis and Design**

**5.1 Sequnce Diagram**

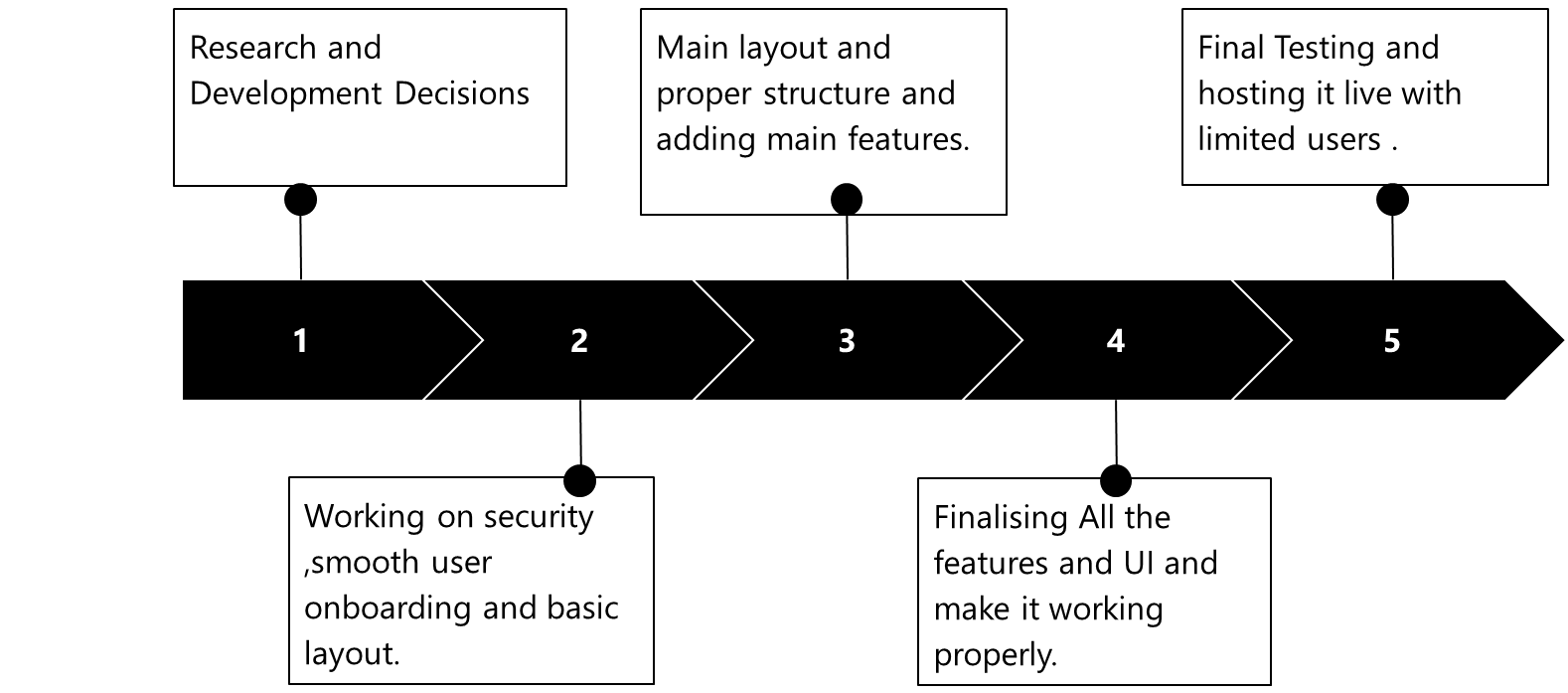
****

16

**5.2 DFD of the Project**

****

**5.3 Gantt chart**

****

17

**6. Implementation**

##### Used Algorithms/Approaches for Project

**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 serverside 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 18

##### 6.2 Implementation of Modules/Algorithms

The modules used for our project includes :

* Multer for uploading : Multer is a node.js middleware for handling multipart/form-data, which is primarily used for uploading files.

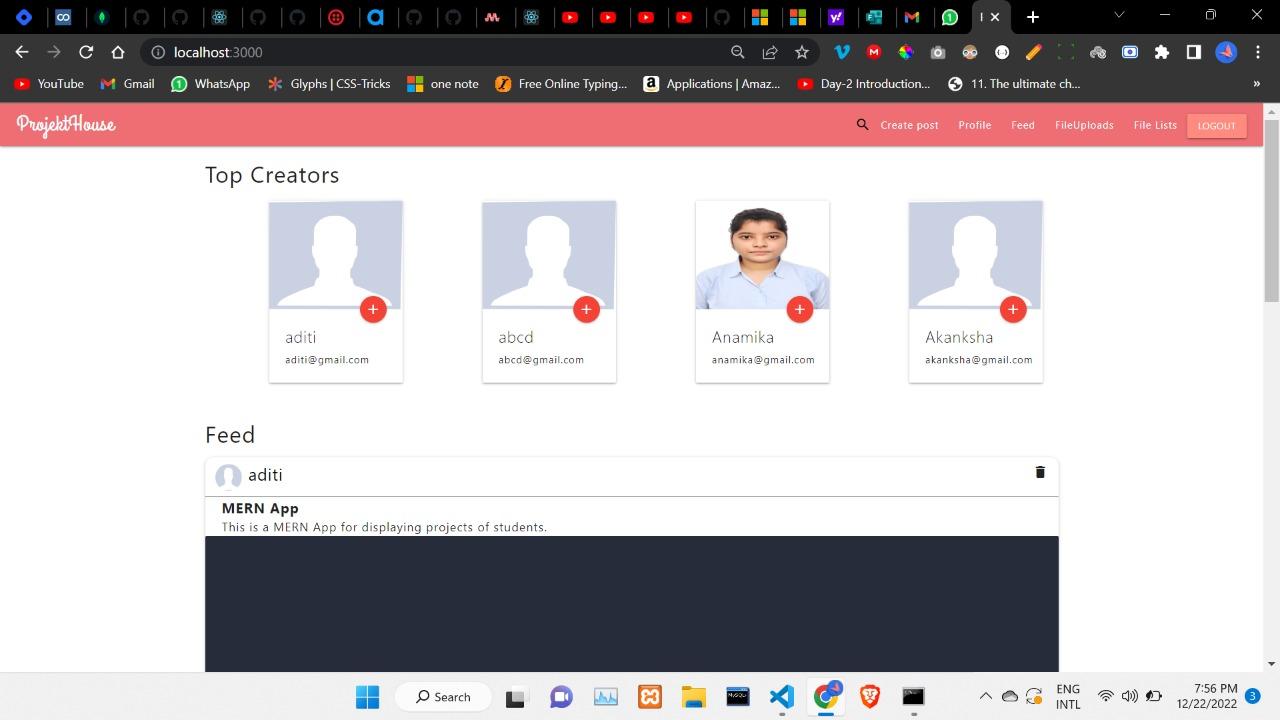
Multer adds a body object and a file or files object to the request object. The body object contains the values of the text fields of the form, the file or files object contains the files uploaded via the form.

* MongoDB for database : MongoDB is a document database. It stores data in a type of JSON format called BSON.A record in MongoDB is a document, which is a data structure composed of key value pairs similar to the structure of JSON objects.
* React for User Interface : React is a JavaScript library for building user interfaces. It is used to build single-page applications. It allows us to create reusable UI components.
* Express for Backend : Express is a minimal and flexible **Node**.**js** web application framework that provides a robust set of features for web and mobile applications.
* Node for Backend : Node.js is an open source server environment. It allows you to run JavaScript on the server .

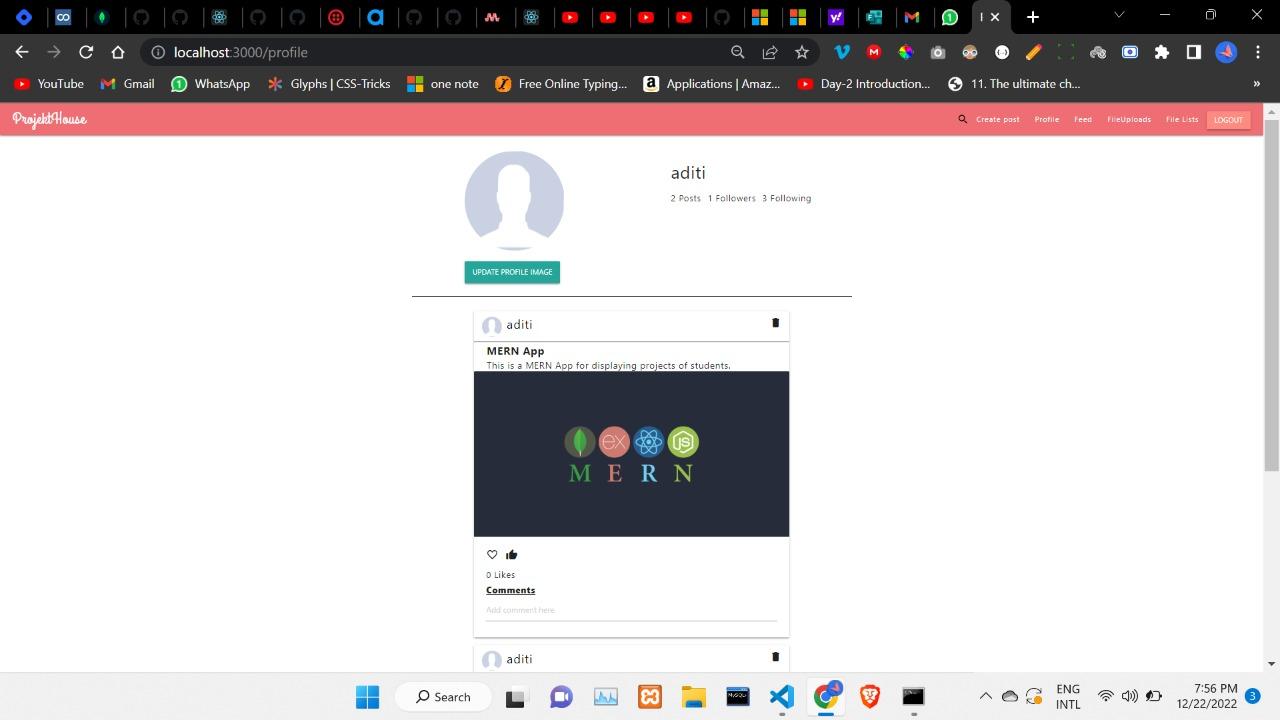
19

**7. Results/Outputs and Testing**

* **Top Creators**

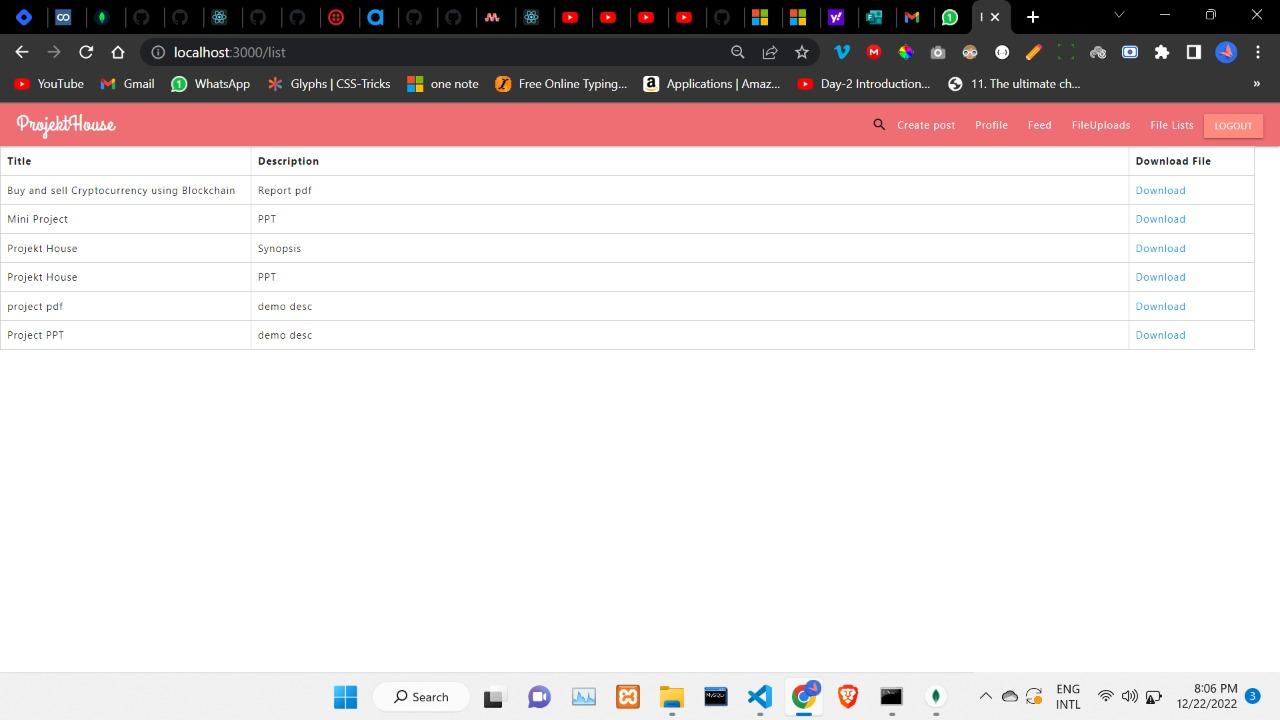
****

* **Profile**

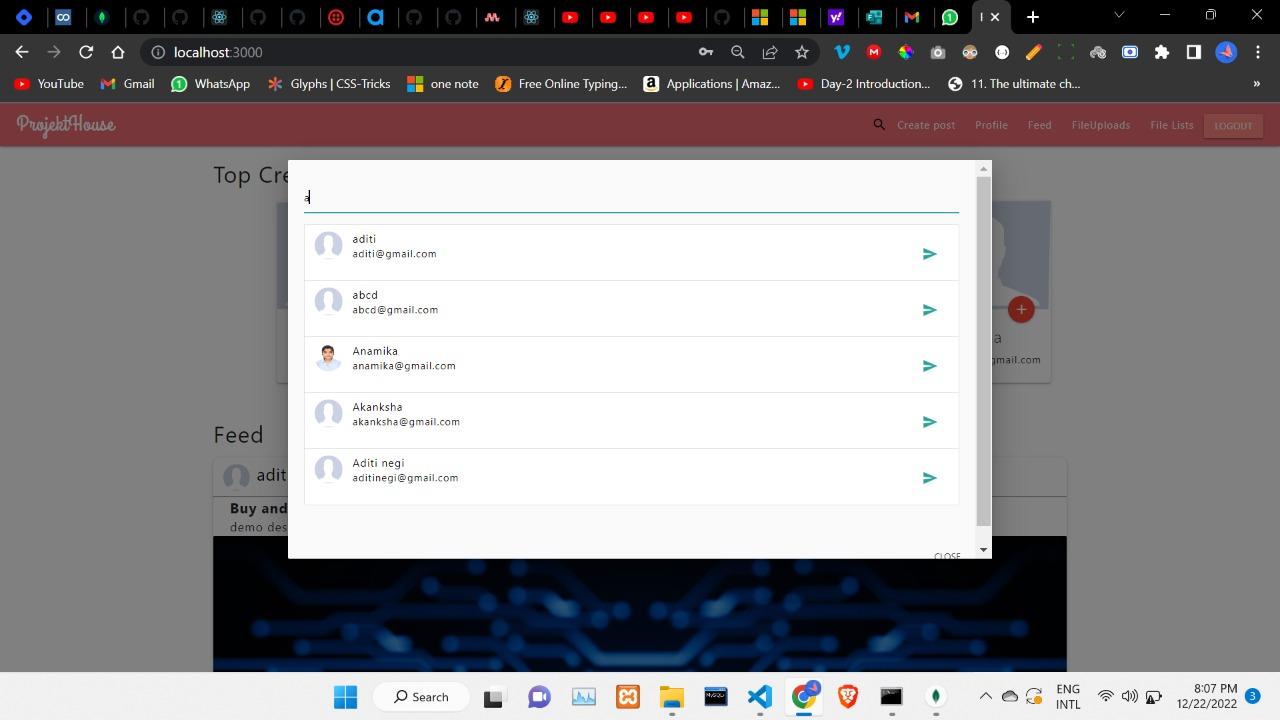
****

20

* **List**

****

* **Search User**

****

21

**8. Conclusions**

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 used by 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 .

22

**9. References**

* 1. Dominik Ertl TU Wien, Harald Krapfenbauer, “A case study of developing an IDE for embedded software using open source,”
  2. Rohit Tamrakar, [Niraj Wani](https://www.researchgate.net/profile/Niraj-Wani), “Design and Development of CHATBOT: A Review,” This paper reviews the technique, terminology, and different platforms used to design and develop the CHATBOT.
  3. Mohammed Farik, Nilesh A. Lal, Shailendra Prasad,“Review Of Authentication Methods,” This Paper describes the importance and types of Authentication mechanism
  4. Hau Tran, “Developing a social platform based on MERN stack” The main goal of this research paper was to study the basic components of the highly popular MERN stack, 2021.
  5. Dominik Ertl TU Wien, Harald Krapfenbauer, “ A Case Study of Developing an IDE for Embedded Software Using Open Source,” This paper recommends developing a customized IDE based on open source software.
  6. Jill Freyne,Shlomo Berkovsky , Elizabeth M. daly, Werner Geyer, “Social networking feeds:Recommending items of interest,” This Research Paper gives the detailed description about the working of feeds for social media platforms .

23

**10. Appendices**

**10.1 Steps to run/execute the project**

* Download and Install all the required libraries and packages
* Navigate to the Downloaded repository and enter : npm install and npm update
* 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 .

**10.2 Coding**

**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, 24

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");

25

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);

});

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) { 26

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" });

* **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 27

(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() { 28

const [state, dispatch] = useReducer(reducer, initialState); return (

<UserContext.Provider value={{ state, dispatch }}>

<BrowserRouter>

<Navbar />

<Routing />

</BrowserRouter>

</UserContext.Provider>

);

}

export default App;

29