API Endpoints Documentation

Repository: api-endpoints

File: README.md

This is an express application that deals with user authentication management. The workflow of the application can be described as follows:

Signup endpoints:

The signup endpoint is built on post request. Users have to enter the following for the signup process:
name, mobile number, email, and password.
The data should be in the form of a JSON object.

After the successful signup process, A JSON toke will be sent along with data whether the signup is successful.

Test case:
{ "name":"anand kumar",
"email":"anandk@gmail.com",
"mobile":"4785123654",
"password":"abcd123@#",
}

Output:

{
"success": true,
"jwt_data":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoiNjNjYzBjOTA2ZjgyZTMyNjEwZjMyNjhlIiwibW9iaWxlIjoiNDc4NTEyMzY1NCIsImVtYWlsIjoiYW5hbmRrQGdtYWlsLmNvbSJ9LCJpYXQiOjE2NzQzMTY5NDR9.lDYTBlpHUoaZHKTKEWNODtpJfzvlSU2RJE_JlVH6DTg"
}

However, if the user already exists then the return value will be an error message which says that the user already exists.

On hitting the endpoint a second time, we will get the following message:

{
"success": false,
"error": "Sorry the user already exists"
}

Login endpoint:
This endpoint also uses the post request.
The endpoint accepts the email and the user's password as the parameters. They should be in JSON format.
If the user enters the correct password corresponding to the email, then a auth token will be sent containing the details such as id, mobile, email, etc.

Test case.

{
"email":"anandk@gmail.com",
"password":"abcd123@#",
}

Output:

{
"success": true,
"authtoken":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoiNjNjYzBjOTA2ZjgyZTMyNjEwZjMyNjhlIiwiZW1haWwiOiJhbmFuZGtAZ21haWwuY29tIiwibW9iaWxlIjoiNDc4NTEyMzY1NCJ9LCJpYXQiOjE2NzQzMTc0MTN9.5Q1UE2KWIUYumn9ryZXJQLK4PzX0crIjacZrUduHqbA"
}

However, an error message will be displayed on entering the incorrect password.

{
"success": false,
"errors": []
}

Reset password:

The reset password also uses post request. The endpoint accepts the email, old password, and new password as the parameters. On entering the correct old password, the user can update the password, and a message containing the new password will be displayed in an encrypted format.

Test case:
{
"email":"anandk@gmail.com",,
"oldPassword":"abcd123@#",
"newPassword":"abcd123@#",
}

Output:

{
"message": "Password reset successfully and new password in hashed forrm is:$2a$10$.SRE6mklo6sjaT8ixnwfp.kP9YpiEQ4pzmQj3RDOr8wmiFJZ15vja"
}

However, if you enter an incorrect old password, then the following message will be shown:

{
"message": "Old password is incorrect."
}

update endpoint:

This endpoint uses the update request. The endpoint accepts name, email, mobile, etc. as

the parameters. One can update the data. The data should be in the form of a JSON object.

Test case:

```
{ "name":"anand kumar",
"email":"anandk@gmail.com",
"mobile":"4785123657",
"password":"abcd12f3@#",
}
```

Output:

```
{
"message": "Data update successfully."
}
```

All the data are encrypted. The private and public keys are auto-generated and can be obtained by running the following commands:

```
var publicKey=key.exportKey('public');
var privateKey=key.exportKey('private');
```

File: db.js

```
const mongoose=require('mongoose');
require('dotenv').config();
const mongoURI=process.env.CONNECTION_URL

const connectToMongoose=async ()=>{
mongoose.connect(mongoURI, async()=>{
console.log('Connected to mongoose successfully.');
})
}

module.exports = connectToMongoose
```

File: index.js

```
const connectToMongoose=require('./db');
const express=require('express');
const app = express();
var cors=require('cors');

const port=process.env.PORT || 5000;

app.use(cors())
app.use(express.json());
app.use('/api/auth',require('./routes/auth'));
```

```
app.get("/",(req,res)=>{
res.send("Hello world and nice to meet you");
})

app.listen(port,()=>{
console.log(`Example app is listening in the port: ${port}`);
})

connectToMongoose();
```

File: fetchuser.js

```
const jwt=require('jsonwebtoken');
const jwt_token = "Hello world";

const fetchuser=async (req,res,next)=>{
const token=await req.header('auth-token');
if(!token){
// res.status(401).send({error:"Please authenticate using a valid token"});
console.log("ok");
}
try{
const data=jwt.verify(token,jwt_token);
req.user=await data.user;
next();
}
catch(err){
res.status(401).send({error:"Please authenticate using a valid token"});
}
}
module.exports = fetchuser
```

File: User.js

```
const mongoose=require('mongoose');
const {Schema}=mongoose;

const UserSchema=new Schema({
name:{
type:String,
required:true
},
mobile:{
type:String,
required:true
},
email:{
type:String,
required:true
},
password:{
type:String,
```

```
required:true
},
date:{
type:Date,
default:Date.now
}
});

const User=mongoose.model('user-assignemnt',UserSchema);
module.exports =User;
```

Repository: asifrahamanportfolio

File: README.md

```
# Welcome to React Ecommerce Webstie Series

## Follow the 3 Steps and your good to go.

1: Clone the Project

2: simply run the command
### `npm install`
First to install all the packages

3: run the project using
### `npm start`

---

# Getting Started with Create React App

This project was bootstrapped with [Create React App](https://github.com/facebook/create-react-app).

## Available Scripts

In the project directory, you can run:

### `npm start`

Runs the app in the development mode.\
Open [http://localhost:3000](http://localhost:3000) to view it in your browser.

The page will reload when you make changes.\
You may also see any lint errors in the console.

### `npm test`

Launches the test runner in the interactive watch mode.\
See the section about [running tests](https://facebook.github.io/create-react-app/docs/running-tests) for more information.

### `npm run build`

Builds the app for production to the `build` folder.\
It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.\
Your app is ready to be deployed!
```

See the section about [deployment](https://facebook.github.io/create-react-app/docs/deployment) for more information.

`npm run eject`

Note: this is a one-way operation. Once you `eject`, you can't go back!

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

Learn More

You can learn more in the [Create React App documentation](https://facebook.github.io/create-react-app/docs/getting-started).

To learn React, check out the [React documentation](https://reactjs.org/).

Code Splitting

This section has moved here: https://facebook.github.io/create-react-app/docs/code-splitting

Analyzing the Bundle Size

This section has moved here: https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size

Making a Progressive Web App

This section has moved here: https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app

Advanced Configuration

This section has moved here: https://facebook.github.io/create-react-app/docs/advanced-configuration

Deployment

This section has moved here: https://facebook.github.io/create-react-app/docs/deployment

`npm run build` fails to minify

This section has moved here: https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify

L1QW5efWQIwxsaBqtsSng7o5Cj9gmAaT5zSR2aFa0z2C9ocV

File: index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta name="theme-color" content="#000000" />
<meta
name="description"
content="Web site created using create-react-app"
/>
<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
<!--
manifest.json provides metadata used when your web app is installed on a
user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
-->
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<!--
Notice the use of %PUBLIC_URL% in the tags above.
It will be replaced with the URL of the `public` folder during the build.
Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
work correctly both with client-side routing and a non-root public URL.
Learn how to configure a non-root public URL by running `npm run build`.
-->
<title>Asif Rahaman Portfolio</title>

</head>
<body>
<noscript>You need to enable JavaScript to run this app.</noscript>
<div id="root"></div>
<script src="https://kit.fontawesome.com/3d050caad7.js"
crossorigin="anonymous"></script>
```

```
<!--
This HTML file is a template.
If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.
The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.
To create a production bundle, use `npm run build` or `yarn build`.
-->

</body>
</html>
```

File: robots.txt

```
# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
```

File: App.css

```
body {
display: flex;
flex-direction: column;
height: 100vh;
overflow-x: hidden;
scroll-behavior: smooth !important;
}

html {
overflow-x: hidden;
overflow-y: hidden;

}

*{
margin: 0px;
}
```

File: App.js

```
import React from "react";
import Navbar from "./components/Navbar/Navbar";
import About from "./components/About/About";
import Contact from "./components/Contact/Contact";
import Home from "./components/Home/Home";
import Footer from "./components/Footer/Footer";
```

```
import "./App.css"
import { BrowserRouter, Routes, Route } from "react-router-dom";
const url="https://portfoliodummy.adaptable.app/"

const App = () => {
return (
<>
<>
<body>
<BrowserRouter>
<Navbar />
<Routes>
<Route path="/asifrahamanportfolio" element={<Home />} />
<Route path="/about" element={<About />} />
<Route path="/contact" element={<Contact />} />
<Route path="*" element={<Home/>} />
</Routes>
<Footer />
</BrowserRouter>
</body>
</>

</>
);
};

export default App;
```

File: About.css

```
.about-container {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
padding: 5% 5% 5% 5%;
margin: 10vh 0vw 0vw 0vw;
}

.about {
text-align: justify;
box-shadow: 0px 5px 20px rgb(153, 135, 212);
padding: 3% 5% 5% 5%;
border-radius: 20px;
width:70%;

}

aboutme {
font-size: 32px;
font-weight: bold;
color: rgb(74, 7, 151);
```

```
}
@media (max-width: 800px) {
.about-container {
margin: 30vh 0vw 10vh 0vh;
}
.about{
width:95%;
}
}
```

File: About.jsx

```
import React from "react";
import "./About.css";

const About = () => {
return (
<>
<div className="about-container">
<div className="about">
<center>
<aboutme>About Me</aboutme>
</center>
<br />
Hello visitor! Thanks for visiting my website. My name is Asif
Rahaman, and I am a second-year student at IIT Bhilai. I am pursuing
Electrical Engineering. Currently, I am in my second year. I am a
coding geek who likes to spend most of my time coding. I code for fun
and to learn the latest amazing technologies. I passed class 10 from
LMET International School and class 12 from Prabharani Public School.
Both of them are in Berhmapore, and both of them come under CBSE.
<p />
<br />
I spend a good amount of my time reading books. I read books from
almost all kinds of domains. Among all horror, my favorites are
science fiction, psychological thrillers, and adventures. I never miss
the stories of Sunday suspense, and I frequently listen to the stories
by radio 98.4 Mirchi Bangle.(I am from West Bengal). I read spiritual
books, too, like Quran, Mahabharat, the bible, etc. This gives me
inner satisfaction and boosts my confidence. I am an agonist, and I
sincerely thank GOD for all his creation.:)
<p />
<br />I love playing sports. I occasionally play football, cricket,
tennis, etc. Among all, football is my favorite. However, due to my
busy schedule, I am not able to play them:(
</div>
</div>
</>
);
};

export default About;
```

File: Banner.css

```
.banner-img{
background-image: url("../assests/19362653.jpg");
background-size: 100% 100%;
width:100vw;
height: 120vh;
}

.banner{
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
}

@media (max-width:500px){
.banner-img{
background-size: 100% 100%;
width:100vw;
margin:25vh 0px 0px 0px;
height: 50vh;

}
.banner{
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;

}
}
```

File: Banner.jsx

```
import React from 'react'
import "./Banner.css"

const Banner = () => {
return (
<>
<div className="banner">
<div className="banner-img"></div>
</div>
</>
)
}

export default Banner
```

File: Bouncing.css

```
bouncing {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
height: 20vh;
}

.bouncing-ul {
display: flex;
flex-direction: row;
}

.bouncing-ul li {
width: 40px;
height: 40px;
list-style: none;
background-color: rgb(212, 163, 224);
margin: auto 2vw;
border-radius: 50px;
animation: bouncing 1s linear infinite alternate-reverse;
}

@keyframes bouncing {
0% {
transform: translateY(0);
}
50% {
transform: translateY(-15vh);
}
100% {
transform: translateY(0);
}
}

.bouncing-ul li:nth-child(2) {
animation-delay: 0.2s;
}
.bouncing-ul li:nth-child(3) {
animation-delay: 0.4s;
}
.bouncing-ul li:nth-child(4) {
animation-delay: 0.8s;
}
.bouncing-ul li:nth-child(5) {
animation-delay: 0.2s;
}
```

File: Bouncing.jsx

```
import React from 'react'
import "./Bouncing.css"

const Bouncing = () => {
return (
<>
<div className="ball">
<bouncing>
<ul className="bouncing-ul">
<li className="bouncing-li"></li>
<li className="bouncing-li"></li>
<li className="bouncing-li"></li>
<li className="bouncing-li"></li>
<li className="bouncing-li"></li>
</ul>
</bouncing>
</div>
</>
)
}

export default Bouncing
```

File: Contact.css

```
.contact-container {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
margin: 10vh;
}

.form-input {
border-color: white;
margin: 1%;
display: flex;
align-items: center;
flex-direction: column;
width: 70vw;
height: 7vh;
padding: 4%;
border-radius: 30px;
box-shadow: 0px 5px 20px rgb(197, 191, 220);
}

.textarea-contact {
margin: 1%;
display: flex;
```

```
align-items: center;
flex-direction: column;
padding: 4%;
width: 70vw;
height: 30vh;
border-radius: 30px;
box-shadow: 0px 5px 20px rgb(197, 191, 220);
}

.contact-me {
font-size: 32px;
font-weight: 700;
color: rgb(26, 1, 55) !important;
}

.btn-submit {
background-color: rgb(74, 7, 151);
color: white;
font-size: 2rem;
font-weight: bold;
width: 20vw;
height: 10vh;
border-radius: 30px;
}

@media (max-width: 500px) {
.btn-submit {
background-color: rgb(74, 7, 151);
color: white;
font-size: 80%;
font-weight: bold;
width: 35vw;
border-radius: 30px;
}
.contact-me {
font-size: 130%;
font-weight: 700;
color: rgb(47, 5, 94) !important;
}
.contact-container {
margin: 20vh 0vw 10vw 0vw;
}
}

#contact-me {
font-size: 50px;
}

.contact-label {
color: rgb(124, 38, 161);
}
.status{
font-size: 1rem;
color:green;
```

```
}
```

File: Contact.jsx

```
import React from "react";
import "./Contact.css";
import { useState } from "react";

const Contact = () => {
const [msg, setMsg] = useState(false);

const [credential, setCredential] = useState({
email: "",
name: "",
});

const handleSubmit = async (e) => {
e.preventDefault();
const response = await fetch(
"https://backendportfolio.adaptable.app//contacts",
{
method: "POST",
headers: {
"Content-Type": "application/json",
},
body: JSON.stringify({
email: credential.email,
name: credential.name,
}),
}
);
const json = await response.json();
console.log(json);
};

const onChange = (e) => {
setCredential({ ...credential, [e.target.name]: e.target.value });
};

return (
<>

<center>
<contact className="contact-header" id="contact-me">
CONTACT ME
</contact>

<div className="contact-container">
<div className="contact-me">
<form onSubmit={handleSubmit}>
<label htmlFor="" className="contact-label">
Enter the name
</label>
```

```
<input
type="text"
className="form-input"
placeholder="Enter your name e.g Captain Price"
name="name"
value={credential.name}
onChange={onChange}
/>
<label htmlFor="" className="contact-label">
Enter your email address
</label>
<input
type="text"
className="form-input"
placeholder="Enter your e-mail address"
name="email"
value={credential.email}
onChange={onChange}
/>
<label htmlFor="" className="contact-label">
Enter your profession
</label>
<input
type="text"
className="form-input"
placeholder="Enter your profession e.g student"
/>
<label htmlFor="" className="contact-label">
Enter your age
</label>
<input
type="text"
className="form-input"
placeholder="Enter your age"
/>
<label htmlFor="" className="contact-label">
Enter your concern
</label>
<textarea
name="textarea-contact"
cols="80"
rows="10"
className="textarea-contact"
id="textarea-contact"
placeholder="Enter your concern e.g Hello can you please.... "
></textarea>
<center>
<button
className="btn-submit"
onClick={(e) => {
setMsg(true);
}}
>
Submit
```

```
</button>
</center>
<submission className="status">
{msg == false ? <></> : <>Your message is submitted</>}
</submission>
</form>
</div>
</div>
</center>
</>
);
};

export default Contact;
```

File: Credits.css

```
.credit-serv ul {
display: flex;
flex-wrap: wrap;
padding-left: 0;
flex-direction: row;
justify-content: center;
}

.credit-serv ul li {
list-style: none;
flex: 5 2 5%;
margin: 1px;
border-radius: 20px;
padding: 1vw 2vw 1vw 2vw;
text-align: center;
box-shadow: 0px 5px 20px rgb(200, 190, 232);
background-color: yellow;
}

.credit-container {
display: flex;
flex-direction: row;
flex-wrap: wrap;
align-items: center;
justify-content: center;
margin:10vh;
}

.skill{
margin: 10vh 0vh 0vh 0vh;
}
```

File: Credits.jsx

```
import React, { useState, useEffect } from "react";
import { ethers } from "ethers";
import "./Credits.css";
import { v4 } from 'uuid'

const Credits = () => {
const data = require("./credits.json");
useEffect(() => {
main();
}, []);

const [creditdata, setCreditdata] = useState([]);

const main = async () => {
setCreditdata(await contract.getCredits());
return;
};
const provider = new ethers.providers.JsonRpcProvider(
`https://goerli.infura.io/v3/5f1919e74ef0420ca8348dfab3af6bdc`
);
const contractAddress = "0x8B7D297a15844415AC7F51fe90C659006F18de2b";
const ABI = data;
const contract = new ethers.Contract(contractAddress, ABI, provider);

return (
<>
<center>
<h2 className="skill" id="credits">
<a href="#scroll">CREDITS</a>
</h2>
</center>
<div className="credit-container">
<div className="credit-serv">
<ul>
{creditdata.map((item, idx) => {
return (
<>
<li key={v4()}> {item}</li>
</>
);
})}
</ul>
</div>
</div>
</>
);
};

export default Credits;
```

File: Footer.css

```
/* .footer-components {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
}

@media (max-width: 500px) {
.footer-components {
flex-wrap: wrap;
}
}
.footer {
background-color: rgb(249, 247, 255);
height: 1200px !important;
margin-top: auto;
}
.footer-components * {
width: 4vw;
}

.github {
background-image: url("../assests/github.png");
background-size: 100% 100%;
}
.facebook {
background-image: url("../assests/facebook.png");
background-size: 100% 100%;
}
.instagram {
background-image: url("../assests/instagram.png");
background-size: 100% 100%;
}
.linkedin {
background-image: url("../assests/linkedin.png");
background-size: 100% 100%;
}
.telegram {
background-image: url("../assests/telegram.png");
background-size: 100% 100%;
}
.gmail {
background-image: url("../assests/gmail.png");
background-size: 100% 100%;
}

.skills {
color: rgb(39, 3, 80);
}

a {
text-decoration: none;
}
```

```
@media (max-width: 500px) {
.footer-components * {
display: flex !important;
flex-direction: row !important;
flex-wrap: wrap !important;
}
}

@media ((max-width: 800px) and (min-width:400px)) {
.footer * {
width: 25vw;
height: 15vh;
margin: 1vh;
}
.footer {
display: flex;
flex-direction: column;

align-items: center;
height: 1000px;
}
.footer-components {
display: flex;
flex-direction: row !important;
flex-wrap: wrap;
margin: 1px;
height: 100vh;
}
.footer-components * {
display: flex;
flex-direction: row !important;
flex-wrap: wrap;
margin: 0.1px;
}
} */
.footer-container{
display:flex;
flex-direction: column;
justify-content: center;
align-items: center;
background-color: rgb(219, 199, 250);
height:1000px !important;
}

.footer-ul-second{
display:flex;
flex-direction: row;
flex-wrap: wrap;
justify-content: center;
align-items: center;
height:40vh !important;
}

.footer-ul-first{
```

```
margin:2vh;
font-size: 2rem;
font-weight: bold;
color: rgb(78, 30, 136);
}

.footer-ul-second li{
padding:2vw;
margin:3vw;
height:2vh;
width:2vh;
}

@media (max-width:800px){
.footer-ul-second li{
padding:2vw;
margin:3vw;
height:5vh;
width:5vh;
}
}
```

File: Footer.jsx

```
import React from "react";
import "./Footer.css";

const Footer = () => {
const images = [
{
handle: "images/linkedin.png",
link: "https://www.linkedin.com/in/asif-rahaman-110099229/",
},
{
handle: "images/telegram.png",
link: "https://telegram.me/asifrahaman_13",
},
{
handle: "images/discord.png",
link: "http://discordapp.com/users/1025373541056647248",
},
{
handle: "images/gmail.png",
link: "https://mail.google.com/mail/?view=cm&fs=1&to=asifrahaman162@gmail.com",
},
{
handle: "images/github.png",
link: "https://github.com/asifrahaman13",
},
];
```

```
const footerstyle1 = {
backgroundSize: "100% 100%",
};

return (
<>
<div className="footer-container">
<ul className="footer-ul-first">CONTACT ME</ul>
<ul className="footer-ul-second">
{images.map((item, idx) => {
return (
<>
<a href={item.link}>
<li
style={{ backgroundImage: `url(${item.handle})`, ...footerstyle1 }}
>
{" "}
</li>
</a>
</>
);
})}
</ul>
</div>
</>
);
};

export default Footer;
```

File: Header.css

```
@media (max-width: 500px) {
.imageclass * {
width: 30vw;
height: 15vw;
margin: 0.1vh;
}
.images {
margin: 5vw 0vw 20vw 0vw;
height: 30vw;
}
aboutme {
margin: 15vw 0vw 0vw 0vw;
}
}

.scroll-view {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
```

```
margin: 4vw;
}

@media (max-width: 500px) {
.scroll-view * {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
flex-wrap: wrap;
padding: 2vw;
font-size: 2vh;
}
}

.scroll-view * {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
flex-wrap: wrap;
padding: 2vw;
font-size: 2vh;
border-color: black;
}

.scroll-child li * {
padding-bottom: 1vh;
border-bottom-style: solid;
border-bottom-width: 0.3vh;
width: max-content;
border-color: rgb(94, 44, 144);
animation: header 5s linear;
}

@keyframes header {
0% {
margin: 100px;
}
20% {
margin: 80px;
}
40% {
margin: 60px;
}
60% {
margin: 40px;
}
}

.hobbies {
font-size: 1rem !important;
}
```

```
.hobbies-container {
display: flex;
flex-direction: row;
align-items: center !important;
justify-content: center !important;
width: 60vw !important;
background-color: red;
}

.txt {
font-size: 1rem !important;
}
```

File: Header.jsx

```
import React from "react";
import "./Header.css";

const Header = () => {
return (
<>
<br />
<br />
<div className="scroll-view" id="scroll">
<div className="scroll-child">
<li>
<a href="#participations">Participations</a>
</li>
<li>
<a href="#projects">Projects</a>
</li>
<li>
<a href="#skills">My Skills</a>
</li>
<li>
<a href="#proficiency">Proficiency</a>
</li>
<li>
<a href="#credits">Credits</a>
</li>
</div>
</div>
<aboutme>
<center>
<h2 className="skill" id="about">
<a href="#nav">MY HOBBIES</a>
</h2>
<br />
<br />
<center>

<h3 className="txt">
```

```
I am a coding geek and a keen learner. I love exploring about new technologies. Apart from that I also spend my time in reading books every day.Ocassionally I listen to sunday suspense....
</h3></center>
</center>
<div className="images">
<div className="imageclass">
<div className="first-image"></div>
<div className="second-image"></div>
<div className="third-image"></div>
</div>
</div>
</aboutme>
</>
);
};

export default Header;
```

File: Home.css

```
.serv ul {
display: flex;
flex-wrap: wrap;
padding-left: 0;
flex-direction: row;
justify-content: center;
}

.serv ul li {
list-style: none;
flex: 0 0 30.333333%;
margin: 1px;
border-radius: 20px;
padding: 2%;
box-shadow: 0px 5px 20px rgb(200, 190, 232);
}

.home-container {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
}

* {
margin: 0;
box-sizing: border-box;
padding: 0px;
}
html {
scroll-behavior: smooth;
```

```
}
.ul {
display: flex;
flex-direction: row-reverse;
list-style: none;
}

.list {
padding: 30px;
color: rgb(81, 31, 180);
font-size: 16px;
}
.nav {
background-color: rgb(249, 247, 255);
height: 700px;
}

.img {
width: 100vw;
display: flex;
flex-direction: row;
justify-content: center;
}

.body-ul {
display: flex;
flex-direction: row;
list-style: none;
width: 70%;
padding: 50px;
}

.body-li {
padding: 50px;
margin: 20px;
border-radius: 15px;
box-shadow: 0px 5px 20px rgb(200, 190, 232);
}
.body {
display: flex;
justify-content: center;
padding: 50px;
height: 100vh;
display: flex;
flex-direction: column;
}

.font {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
height: 50vh;
}
```

```
.first {
padding: 50px;
}
#first {
background-color: rgb(249, 247, 255);
height: 30vh;
display: flex;
flex-direction: row;
justify-content: center;
align-items: center;
}
#second {
background-color: rgb(249, 247, 255);
width: 50vw;
height: 55vh;
display: flex;
flex-direction: row;
justify-content: center;
align-items: center;
background-image: url("../assests/img.png");
background-size: 100% 100%;
}
.btn {
height: 50px;
width: 150px;
border-radius: 30px;
background-color: blueviolet;
box-shadow: 0px 5px 20px rgb(31, 69, 146);
color: white;
font-weight: bold;
font-size: 14px;
justify-content: center;
align-items: center;
}
.flex-box {
display: flex;
flex-direction: row;
text-align: center;
justify-content: center;
align-items: center;
height: 50vh;
margin: 20px;
}
.display {
align-items: center;
padding: 50px;
}
.flex-box * {
border-radius: 10%;
display: flex;
justify-content: justify;
align-items: center;
}
.flex-box1 {
```

```
background-color: red;
width: 45%;
height: 50vh;
margin: 2%;
}
.flex-box2 {
background-color: green;
width: 45%;
height: 50vh;
margin: 2%;
}
.flex-box3 {
background-color: blue;
width: 50%;
height: 50vh;
margin: 2%;
}
#flex1 {
background-color: rgb(249, 247, 255);
}
#flex2 {
background-color: rgb(249, 247, 255);
background-image: url("https://dlcdnrog.asus.com/rog/media/1640906694290.webp");
background-size: 100% 100%;
}
#flex3 {
background-color: green !important;
background-image:
url("https://www.hardsoftcomputers.co.uk/wp-content/uploads/2022/02/How-long-do-gaming-l
aptops-last.jpg");
background-size: 100% 100%;
}
#flex4 {
background-color: rgb(249, 247, 255);
}
#flex5 {
background-color: rgb(249, 247, 255);
}
#flex6 {
background-color: orange !important;
background-image:
url("https://imgnew.outlookindia.com/uploadimage/library/16_9/16_9_5/IMAGE_1651057728.we
bp");
background-size: 100% 100%;
}

.container2 {
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;
margin: 5px 5px 5px 5px;
padding: 4%;
}
```

```
.container2 * {
width: 90%;
display: flex;
flex-direction: row;
justify-content: center;
align-items: center;
height: 30vh;
margin: 10px 10px 10px 10px;
border-radius: 20px;
}

.flexbox1 {
display: flex;
flex-direction: row;
justify-content: center;
align-items: center;
}

.flexbox2 {
display: flex;
flex-direction: row;
justify-content: center;
align-items: center;
}
.flexbox1 * {
padding: 50px;
width: 80vw;
box-shadow: 0px 5px 20px rgb(200, 190, 232);
}
.flexbox2 * {
padding: 50px;
box-shadow: 0px 5px 20px rgb(200, 190, 232);
}

.second-img {
border-radius: 300px;
display: flex;
justify-content: center;
align-items: center;
padding: 5%;
}
.second-img * {
border-radius: 15px;
}

.images {
display: flex;
align-items: center;
justify-content: center;
}

.imageclass {
display: flex;
align-items: center;
```

```
justify-content: center;
}

.imageclass * {
width: 30vw;
height: 30vh;
display: flex;
align-items: center;
justify-content: center;
margin: 8%;
padding: 5%;
}

.first-image {
background-image: url("../assests/py.png");
background-size: 100% 100%;
}
.second-image {
background-image: url("../assests/v.jpg");
background-size: 100% 100%;
}
.third-image {
background-image: url("../assests/pro.jpg");
background-size: 100% 100%;
}

.assests {
display: flex;
align-items: center;
justify-content: center;
}


.footer-components {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
}

.footer-components * {
padding: 40px;
margin: 4vw;
width: 4vw;
}

.github {
background-image: url("../assests/github.png");
background-size: 100% 100%;
}
.facebook {
background-image: url("../assests/facebook.png");
background-size: 100% 100%;
}
```

```
.instagram {
background-image: url("../assests/instagram.png");
background-size: 100% 100%;
}
.linkedin {
background-image: url("../assests/linkedin.png");
background-size: 100% 100%;
}
.telegram {
background-image: url("../assests/telegram.png");
background-size: 100% 100%;
}
.gmail {
background-image: url("../assests/gmail.png");
background-size: 100% 100%;
}

.skills {
padding: 0%;
}

a {
text-decoration: none;
}
```

File: Home.jsx

```
import React from "react";
import "./Home.css";
import Header from "../Header/Header";
import Participations from "../Participations/Participations";
import Skills from "../Skills/Skills";
import Proficiency from "../Proficiency/Proficiency";
import Projects from "../Projects/Projects";
import Banner from "../Banner/Banner";
import Bouncing from "../Bouncing/Bouncing";
import Credits from "../Credits/Credits";

const Home = () => {
return (
<>
<Banner />
<Header />
<Bouncing />
<center>
<h2 className="skill" id="about">
<a href="#scroll" id="participations">
PARTICIPATIONS
</a>
</h2>
<br />
<br />
```

```
<br />
</center>
<Participations />
<br />
<br />
<br />
<br />
<br />
<br />

<center>
<h2 className="skill" id="projects">
<a href="#scroll">PROJECTS</a>
</h2>
</center>
<br />
<br />
<br />
<Projects />
<br />
<br />
<br />
<br />
<br />
<br />
<center>
<h2 className="skill" id="skills">
<a href="#scroll">MY SKILLS</a>
</h2>
</center>

<Skills />
<Proficiency />
<Credits/>
</>
);
};

export default Home;
```

File: index.js

```
import React from 'react';
import createRoot from 'react-dom'
import App from './App'

createRoot.render(
<React.StrictMode>
<App />
</React.StrictMode>,
document.getElementById('root')
);
```

Repository: backend-internship-task

File: README.md

````
# Calorie Tracker Application

## Description
The Calorie Tracker is a web application that allows users to track their daily calorie intake by creating and managing entries.

## Setup
1. Clone the repository:
git clone https://github.com/DiveHQ-Octernships/dive-backend-engineering-octernship-asifrahaman13.git
2. Navigate to the project directory:

3. Create a virtual environment:
python -m venv venv
4. Activate the virtual environment:
- For Windows:
```
venv\Scripts\activate
```
- For macOS/Linux:
```
source venv/bin/activate
```
5. Install the required dependencies:
pip install -r requirements.txt
6. Set up the database:
- Modify the database connection settings in `main.py` according to your database configuration.

## Usage
1. Start the application server:
uvicorn main:app --reload
2. Access the application in your web browser at `http://localhost:8000`.

## API Testing with pytest
1. Make sure the application server is running.
2. Activate the virtual environment if not already activated.
3. Run the pytest cases:
pytest



## Example usage through Postman or Thunder Client

Getting started:

URL: http://0.0.0.0:8000/
````

REQUEST TYPE:

GET

RESPONSE:

Hello, world!
****
****
Now first, you need to register before being able to use the app.

URL: http://0.0.0.0:8000/api/register

Request Type: POST

JSON PALOAD:

```
{
"username":"asifxy",
"password":"pass",
"role":"admin",
"daily_calorie_goal":120
}
```

Response:

```
{
"status_code": 202,
"message": "User registered successfully"
}
```

****
****

Next, you need to log in

URL: http://0.0.0.0:8000/api/login

Request Type: POST

JASON PAYLOAD:

```
{
"username":"asifxyz",
"password":"pass"
}
```

Response:

```
{
"message": "Login successful"
}
```

****
****

Next, you can enter the entries.

URL: http://0.0.0.0:8000/api/entries

Request Type: POST

JSON:

```
{ "date": "2023-06-18",
"time": "12:00",
"text": "I Love Apples Oranges",
"calories": 300,
"username":"asifxyz",
"password":"pass",
"role":"admin",
"daily_calorie_goal":120
}
```

Response:

```
{
"message": "Entry created successfully"
}
```

****
****

Next you can get all the entries:

URL: http://0.0.0.0:8000/api/entries

REQUEST TYPE: GET

JSON PAYLOAD:

```
{ "date": "2023-06-18",
"time": "12:00",
"text": "I Love Apples Oranges",
"calories": 300,
"username":"asifxyz",
"password":"pass",
"role":"admin",
"daily_calorie_goal":120
}
```

Response:

```
[
{
"time": "12:00:00",
```

```
"id": 6,
"calories": 500,
"date": "2023-06-18",
"user_id": 2,
"text": "Test entry",
"is_below_goal": true
},
{
"time": "12:00:00",
"id": 7,
"calories": 500,
"date": "2023-06-18",
"user_id": 2,
"text": "Test entry",
"is_below_goal": true
},
{
"time": "12:00:00",
"id": 8,
"calories": 300,
"date": "2023-06-18",
"user_id": 6,
"text": "I Love Apples Oranges",
"is_below_goal": false
}
]
```


****

To update the details, you need to pass the id of the entry as a slug

URL: http://0.0.0.0:8000/api/entries/8

REQUEST TYPE: PUT

JSON PAYLOAD:


```
{ "date": "2023-06-18",
"time": "12:00",
"text": "I Love Apples only",
"calories": 300,
"username":"asifxy",
"password":"pass",
"role":"admin",
"daily_calorie_goal":120
}
```


RESPONSE:


```
{
"message": "Entry updated successfully"
}
```

To delete, we also need to pass the id in the form of a slug

****
****

URL: http://0.0.0.0:8000/api/entries/8

REQUEST TYPE: DELETE]

RESPONSE:

```
{
"message": "Entry deleted successfully"
}
```

We can test similar functionalities for other access rights like amin, users etc.

File: main.py

```
"""_summary_
References:
Nutritionix API Documentation =
https://gist.github.com/mattsilv/6d19997bbdd02cf5337e9d4806b4f464

"""

# Import all the libraries and packages in the code
from fastapi.staticfiles import StaticFiles
from fastapi import FastAPI, HTTPException, Depends, status
from fastapi import FastAPI
import logging
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker,Session
from sqlalchemy.ext.declarative import declarative_base
from fastapi.middleware.cors import CORSMiddleware
from sqlalchemy import Column, Integer, String, Date, Time, Boolean, ForeignKey, func
from sqlalchemy.orm import relationship
from fastapi.security import HTTPBasic,HTTPBasicCredentials
from enum import Enum
from passlib.context import CryptContext
from starlette.responses import Response
from pydantic import BaseModel
import datetime
from typing import Optional
import requests

# Setting  the debugging mode on
logging.basicConfig(level=logging.DEBUG)

# Setting the FastAPI
app = FastAPI()
```

```
# Mount the static files which would contains the HTML files for testing the APIs
# app.mount("/static", StaticFiles(directory="static"), name="static")


headers = {
'Accept-Language': 'en-US,en;q=0.9',
'Connection': 'keep-alive',
'Referer': 'https://trackapi.nutritionix.com/docs/',
'Sec-Fetch-Dest': 'empty',
'Sec-Fetch-Mode': 'cors',
'Sec-Fetch-Site': 'same-origin',
'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/114.0.0.0 Safari/537.36',
'accept': 'application/json',
'sec-ch-ua': '"Not.A/Brand";v="8", "Chromium";v="114", "Google Chrome";v="114"',
'sec-ch-ua-mobile': '?0',
'sec-ch-ua-platform': '"Windows"',
}

# Added multiple origins to remove the cors errors which we may encounter later

origins = [
"http://localhost",
"http://127.0.0.1",
"http://127.0.0.1:8000",
"http://localhost:8000",
"http://localhost:8000/api",
]

# Setting up the parth for the SQLite
SQLALCHEMY_DATABASE_URL = "sqlite:///calories.db"
engine = create_engine(SQLALCHEMY_DATABASE_URL)
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
Base = declarative_base()

# Middleware to pass on the cors error and to check the credentials
app.add_middleware(
CORSMiddleware,
allow_origins=origins,
allow_credentials=True,
allow_methods=["*"],
allow_headers=["*"],
)

# Function to interact with the database

def get_db():
db = SessionLocal()
try:
yield db
finally:
db.close()
```

```
# Define three roles for user, manager and an admin
class Role(str, Enum):
USER = "user"
USER_MANAGER = "user_manager"
ADMIN = "admin"

# Defining the schema for the users.
''' It will contain the username, password, role, daily calorie goad, and the entries.
The id will be genrated automatically'''
class User(Base):
__tablename__ = 'users'

id = Column(Integer, primary_key=True)
username = Column(String(50), unique=True, nullable=False)
password = Column(String(100), nullable=False)
role = Column(String(20), nullable=False)
daily_calorie_goal = Column(Integer, nullable=False)
entries = relationship('Entry', back_populates='user')

# Defining the schema for the entries.

''' The entries should contain the user id, date , time , the text whixh contains the
food, calories, whether it is below the goal and the user.'''

class Entry(Base):
__tablename__ = 'entries'

id = Column(Integer, primary_key=True)
user_id = Column(Integer, ForeignKey('users.id'), nullable=False)
date = Column(Date, nullable=False)
time = Column(Time, nullable=False)
text = Column(String(100), nullable=False)
calories = Column(Integer)
is_below_goal = Column(Boolean)
user = relationship('User', back_populates='entries')

# The following code is used for database schema creation  and configuration. It will
handle authentication as well as password hasing.
Base.metadata.create_all(bind=engine)
security = HTTPBasic()
pwd_context = CryptContext(schemes=["bcrypt"], deprecated="auto")

def verify_password(plain_password, hashed_password):
return pwd_context.verify(plain_password, hashed_password)


# Function to genrerate the hash function
def get_password_hash(password):
return pwd_context.hash(password)

def get_password_hash(password):
return pwd_context.hash(password)
```

```
def authenticate_user(credentials: HTTPBasicCredentials = Depends(security)):
db = SessionLocal()
user = db.query(User).filter_by(username=credentials.username).first()
if not user or not verify_password(credentials.password, user.password):
raise  HTTPException(status_code=status.HTTP_401_UNAUTHORIZED,  detail="Invalid  username
or password")
return user

# Creating the model for the user
class UserLogin(BaseModel):
username: str
password: str

class EntryUpdate(BaseModel):
date: datetime.date
time: datetime.time
text: str
calories: int

class EntryCreate(BaseModel):
date: datetime.date
time: datetime.time
text: str
calories: Optional[int] = None

class UserCreate(BaseModel):
username: str
password: str
role: Role
daily_calorie_goal: int

# API to access the index page of the web application
@app.get("/")
async def index():
response = Response(content="Hello, World!", media_type="text/plain")
return response

# API to register the user
@app.post("/api/register")
async def register(user: UserCreate, db: Session = Depends(get_db)):
hashed_password = get_password_hash(user.password)
db_user = User(
username=user.username,
password=hashed_password,
role=user.role.value,
daily_calorie_goal=user.daily_calorie_goal
)
db.add(db_user)
db.commit()
# Return the status code and the success message.
return {"status_code": 202, "message": "User registered successfully"}

# API to login
@app.post("/api/login")
```

```
async def login(user: UserLogin, db: Session = Depends(get_db)):
db_user = db.query(User).filter_by(username=user.username).first()
if db_user is None or not verify_password(user.password, db_user.password):
raise HTTPException(status_code=status.HTTP_401_UNAUTHORIZED, detail="Invalid username or password")
return {"message": "Login successful"}

# API to get all the entries of the user
@app.get("/api/entries")
async def get_entries(user: User = Depends(authenticate_user), db: Session = Depends(get_db)):
if user.role == "admin":
entries = db.query(Entry).all()
elif user.role == "user_manager":
entries = db.query(Entry).join(User).all()
else:
entries = db.query(Entry).filter_by(user_id=user.id).all()
return entries

# API to post entries
@app.post("/api/entries")
async def create_entry(entry: EntryCreate, user: User = Depends(authenticate_user), db: Session = Depends(get_db)):
if not entry.calories:
calories = fetch_calories_from_api(entry.text)
if calories is None:
raise HTTPException(status_code=status.HTTP_400_BAD_REQUEST, detail="Unable to fetch calories for the entered meal")
entry.calories = calories
total_calories = get_total_calories_for_day(user, entry.date)
if not total_calories:
total_calories = 0
is_below_goal = total_calories + entry.calories < user.daily_calorie_goal

db_entry = Entry(
user_id=user.id,
date=entry.date,
time=entry.time,
text=entry.text,
calories=entry.calories,
is_below_goal=is_below_goal
)
db.add(db_entry)
db.commit()
return {"message": "Entry created successfully"}

# Fetch calories from the api available
def fetch_calories_from_api(meal_text):
data = {"password": "Drive^&*", "email": "asifrahaman162@gmail.com"}
response = requests.post('https://trackapi.nutritionix.com/v2/auth/signin', headers=headers, data=data)
user_jwt = response.json()["x-user-jwt"]
app_id = "effeff0e"
app_key = "a26584f64a498f2f641bb6034dded946"
```

```
headers.update( {
"x-app-id": app_id, "x-app-key": app_key, "x-user-jwt":  user_jwt,
})
params = {
'query': meal_text,
'self': 'true',
'branded': 'true',
'branded_food_name_only': 'false',
'common': 'true',
'common_general': 'true',
'common_grocery': 'true',
'common_restaurant': 'true',
'detailed': 'false',
'claims': 'false',
'taxonomy': 'false',
}
response       =       requests.get('https://trackapi.nutritionix.com/v2/search/instant',
params=params, headers=headers)
if response.status_code == 200:

response_json = response.json()
if "branded" not in response_json or not response_json["branded"]:
return
return int(response.json()["branded"][-1].get('nf_calories'))

# Get the details of the entry through the ID
@app.get("/api/entries/{entry_id}")
async def get_entry(entry_id: int, user: User = Depends(authenticate_user), db: Session
= Depends(get_db)):
# Check if the entry exists
db_entry = db.query(Entry).filter_by(id=entry_id, user_id=user.id).first()
if not db_entry:
raise HTTPException(status_code=status.HTTP_404_NOT_FOUND, detail="Entry not found")

# Return the entry
return {
"id": db_entry.id,
"user_id": db_entry.user_id,
"date": db_entry.date,
"time": db_entry.time,
"text": db_entry.text,
"calories": db_entry.calories,
"is_below_goal": db_entry.is_below_goal
}

# API to update the entry by ID
@app.put("/api/entries/{entry_id}")
async    def    update_entry(entry_id:    int,    entry:    EntryUpdate,    user:    User    =
Depends(authenticate_user), db: Session = Depends(get_db)):
# Check if the entry exists
db_entry = db.query(Entry).filter_by(id=entry_id, user_id=user.id).first()
if not db_entry:
raise HTTPException(status_code=status.HTTP_404_NOT_FOUND, detail="Entry not found")
```

```
# Update the entry with the provided data
db_entry.date = entry.date
db_entry.time = entry.time
db_entry.text = entry.text
db_entry.calories = entry.calories

# Recalculate is_below_goal
total_calories = get_total_calories_for_day(user, db_entry.date)
is_below_goal = total_calories + entry.calories < user.daily_calorie_goal
db_entry.is_below_goal = is_below_goal

db.commit()

return {"message": "Entry updated successfully"}

# Function to calcualate the total calories of the user per day
def get_total_calories_for_day(user, date):
db = SessionLocal()
total_calories      =      db.query(func.sum(Entry.calories)).filter_by(user_id=user.id,
date=date).scalar()
return total_calories

# Update the entries of of the food
@app.put("/api/entries/{entry_id}")
async def update_entry(
entry_id: int,
entry: EntryUpdate,
user: User = Depends(authenticate_user),
db: Session = Depends(get_db),
):
db_entry = db.query(Entry).get(entry_id)
if not db_entry:
raise HTTPException(
status_code=status.HTTP_404_NOT_FOUND,
detail="Entry not found",
)

if user.role != "admin" and db_entry.user_id != user.id:
raise HTTPException(
status_code=status.HTTP_403_FORBIDDEN,
detail="You don't have permission to update this entry",
)

db_entry.date = entry.date
db_entry.time = entry.time
db_entry.text = entry.text

db.commit()
return {"message": "Entry updated successfully"}


# API to delete the entries completely
@app.delete("/api/entries/{entry_id}")
async def delete_entry(
```

```
entry_id: int,
user: User = Depends(authenticate_user),
db: Session = Depends(get_db),
):
db_entry = db.query(Entry).get(entry_id)
if not db_entry:
raise HTTPException(
status_code=status.HTTP_404_NOT_FOUND,
detail="Entry not found",
)

if user.role != "admin" and db_entry.user_id != user.id:
raise HTTPException(
status_code=status.HTTP_403_FORBIDDEN,
detail="You don't have permission to delete this entry",
)

db.delete(db_entry)
db.commit()
return {"message": "Entry deleted successfully"}

# Driver code of the program
if __name__ == "__main__":
import uvicorn
uvicorn.run(app, host="0.0.0.0", port=8000)
```

File: requirements.txt

```
anyio==3.7.0
bcrypt==4.0.1
certifi==2023.5.7
charset-normalizer==3.1.0
click==8.1.3
colorama==0.4.6
dnspython==2.3.0
email-validator==2.0.0.post2
exceptiongroup==1.1.1
fastapi==0.97.0
FastAPI-SQLAlchemy==0.2.1
greenlet==2.0.2
h11==0.14.0
httpcore==0.17.2
httptools==0.5.0
httpx==0.24.1
idna==3.4
iniconfig==2.0.0
itsdangerous==2.1.2
Jinja2==3.1.2
MarkupSafe==2.1.3
orjson==3.9.1
packaging==23.1
passlib==1.7.4
pluggy==1.0.0
```

```
pydantic==1.10.9
pytest==7.3.2
python-dotenv==1.0.0
python-multipart==0.0.6
PyYAML==6.0
requests==2.31.0
sniffio==1.3.0
SQLAlchemy==2.0.16
starlette==0.27.0
tomli==2.0.1
typing_extensions==4.6.3
ujson==5.8.0
urllib3==2.0.3
uvicorn==0.22.0
watchfiles==0.19.0
websockets==11.0.3
```

File: submission.md

Calorie Tracker Application

Description
The Calorie Tracker is a web application that allows users to track their daily calorie intake by creating and managing entries.

Setup
1. Clone the repository:
git clone https://github.com/MaheswaranPalaniselvan/nutritionix_api_fast_api.git
2. Navigate to the project directory:

3. Create a virtual environment:
python -m venv venv
4. Activate the virtual environment:
- For Windows:
```
venv\Scripts\activate
```
- For macOS/Linux:
```
source venv/bin/activate
```
5. Install the required dependencies:
pip install -r requirements.txt
6. Set up the database:
- Modify the database connection settings in `main.py` according to your database configuration.

Usage
1. Start the application server:
uvicorn main:app --reload
2. Access the application in your web browser at `http://localhost:8000`.

API Testing with pytest
1. Make sure the application server is running.
2. Activate the virtual environment if not already activated.
3. Run the pytest cases:
pytest

File: test_main.py

```
import pytest
from fastapi.testclient import TestClient
from main import app
from datetime import date, time
from unittest.mock import Mock, patch
from main import fetch_calories_from_api
all_entries = []
client = TestClient(app)

def test_create_entry():
entry_data = {
"date": str(date.today()),
"time": str(time(12, 0)),
"text": "Test entry",
"calories": 500
}
response = client.post("/api/entries", json=entry_data,auth=('asifxy', 'pass'))
assert response.status_code == 200
assert response.json() == {"message": "Entry created successfully"}

def test_get_entries():
global all_entries
response = client.get("/api/entries",auth=('asifxy', 'pass'))
assert response.status_code == 200
assert isinstance(response.json(), list)
all_entries = [entry["id"] for entry in response.json()]

def test_get_entry():
response = client.get(f"/api/entries/{all_entries[0]}",auth=('asifxy', 'pass'))
assert response.status_code == 200
assert isinstance(response.json(), dict)

def test_update_entry():
updated_entry_data = {
"date": str(date.today()),
"time": str(time(14, 0)),
"text": "Updated entry",
"calories": 600
}
response                  =                  client.put(f"/api/entries/{all_entries[0]}",
json=updated_entry_data,auth=('asifxy', 'pass'))
assert response.status_code == 200
assert response.json() == {"message": "Entry updated successfully"}
```

```
def test_delete_entry():
response = client.delete(f"/api/entries/{all_entries[0]}",auth=('asifxy', 'pass'))
assert response.status_code == 200
assert response.json() == {"message": "Entry deleted successfully"}

def test_create_entry_missing_fields():
entry_data = {}
response = client.post("/api/entries", json=entry_data,auth=('asifxy', 'pass'))
assert response.status_code == 422

def test_get_entry_not_found():
response = client.get("/api/entries/999",auth=('asifxy', 'pass'))
assert response.status_code == 404

def test_update_entry_not_found():
updated_entry_data = {
"date": str(date.today()),
"time": str(time(14, 0)),
"text": "Updated entry",
"calories": 600
}
response    =    client.put("/api/entries/999",    json=updated_entry_data,auth=('asifxy',
'pass'))
assert response.status_code == 404

def test_delete_entry_not_found():
response = client.delete("/api/entries/999",auth=('asifxy', 'pass'))
assert response.status_code == 404


def test_fetch_calories_from_api_successful():
mock_response = Mock()
mock_response.status_code = 200
mock_response.json.return_value = {
"branded": [
{
"nf_calories": 300
}
]
}

with patch('requests.get', return_value=mock_response):
calories = fetch_calories_from_api("meal")
assert calories == 300

def test_fetch_calories_from_api_no_branded_food():
mock_response = Mock()
mock_response.status_code = 200
mock_response.json.return_value = {
"branded": []
}

with patch('requests.get', return_value=mock_response):
calories = fetch_calories_from_api("meal")
```

```
assert calories is None

def test_fetch_calories_from_api_error():
mock_response = Mock()
mock_response.status_code = 500

with patch('requests.get', return_value=mock_response):
calories = fetch_calories_from_api("meal")
assert calories is None
```

Repository: backend-repo

File: data.py

```
import json
import os
from github import Github
from fpdf import FPDF
from dotenv import load_dotenv
import time

# Load environment variables from .env file
load_dotenv()

# This is the acess key of the person who will use this code. Note that this has usage limit and needs to be reset if the limit is reached.
# Also I have set expiration date of 30 days.
github_access_key = os.environ['GITHUB_ACCESS_TOKEN']

def fetch_github(username):
# Initialize an empty list to hold the user information.
userinfo=[]
# Replace 'YOUR_ACCESS_TOKEN' with your personal access token
access_token = github_access_key
# Create a Github instance with the access token
g = Github(access_token)
# Get a user by username
user = g.get_user(username)
userinfo.append(user.name)
userinfo.append(user.login)
userinfo.append(user.location)
userinfo.append(user.bio)
# List repositories of the user
repositories = user.get_repos()
count = 0
# Create an empty repsonse object where we will store the results
response = {}
print("Getting all the information online.....................................................")
# Start the timer
start_time = time.time()
for repo in repositories:
repository = g.get_repo(f'{username}/{repo.name}')
count += 1
if count >= 15:
break
try:
repo_data = {}
traverse_repository(repository, repo_data, 0)
if repo_data:
response[repo.name] = repo_data
except Exception as e:
```

```
print(e)
end_time = time.time()
# Check if the JSON file exists
json_file_path = f"json_data/repository_codes_{username}.json"
if not os.path.exists(json_file_path):
# Create the file if it doesn't exist
open(json_file_path, "w").close()

# Save response as JSON
with open(json_file_path, "w") as json_file:
json.dump(response, json_file, indent=4)
userinfo.append(end_time-start_time)
return userinfo


'''A helper function that traverse the repository. Even if there are folders and nested files the function would traverse each  of them recursively to find the code bases.'''
# In this code I have kept the number of files to be traversed to be 5 only to keep it short and simple. Otherwise it will take good amount of time to traverse the entire repository.
def traverse_repository(repository, repo_data, count, path=""):
IGNORED_FOLDERS=["node_modules", "vs", "venv"]
contents = repository.get_contents(path)
files_traversed = 0  # Track the number of files traversed
should_exit = False  # Flag to determine when to exit the function
while contents:
# Condition to break the loop and eventually the function itself. If you want all the files then remove this line of codes.
if files_traversed >= 8 or count >= 8:
should_exit = True  # Set the flag to exit the function
break
file_content = contents.pop(0)
if file_content.type == "dir":
if file_content.name in IGNORED_FOLDERS:
continue  # Ignore the directory and continue to the next iteration
files_traversed += 1
traverse_repository(repository, repo_data,
count+1, file_content.path)
else:
files_traversed += 1
file_extension = file_content.path.split(".")[-1]
if file_extension in ["txt", "py", "ipynb", "md", "html", "htm", "css", "jsx", "js", 'sol', 'cpp', "kt", "tsx", "ts", "go", "php", "sol","java"]:
file_name = file_content.path.split("/")[-1]
file_data = {
"code": file_content.decoded_content.decode("utf-8")
}
repo_data[file_name] = file_data
tt = file_content.decoded_content.decode("utf-8")
print(f"{file_name} ***********************************")
print(f"{tt}")
# Increment the count of files traversed
if should_exit:
return
```

```
def convert_to_pdf(username):
print("Converting into PDFs.....................................")
# Read the JSON data from the file
with open(f'json_data/repository_codes_{username}.json', 'r') as file:
data = json.load(file)
# Create a PDF document
pdf = FPDF()
# Set up the document
pdf.set_title('API Endpoints Documentation')
pdf.set_auto_page_break(auto=True, margin=15)
# Add a cover page
pdf.add_page()
pdf.set_font('Arial', 'B', 24)
pdf.cell(0, 20, 'API Endpoints Documentation', ln=True, align='C')
# Iterate over the JSON data and add repository name and files with code to the PDF
pdf.set_font('Arial', 'B', 16)
for repository, files in data.items():
pdf.add_page()
pdf.set_font('Arial', 'B', 14)
# Handle Unicode characters in repository name
repository = repository.encode('latin-1', 'replace').decode('latin-1')
pdf.cell(0, 10, f'Repository: {repository}', ln=True, align='L')
pdf.set_font('Arial', '', 12)
for filename, content in files.items():
pdf.multi_cell(0, 10, f'File: {filename}')
pdf.set_font('Courier', '', 10)
# Handle Unicode characters in code content
code = content['code'].encode(
'latin-1', 'replace').decode('latin-1')
pdf.multi_cell(0, 10, code)
# Save the PDF document with UTF-8 encoding
pdf.output(f'pdf_data/api_endpoints_{username}.pdf', 'F')


def convert_to_formatted_pdf(username):
print("Converting into PDFs.....................................")
# Read the JSON data from the file
with open(f'json_data/repository_codes_{username}.json', 'r') as file:
data = json.load(file)
# Create a PDF document
pdf = FPDF()
# Set up the document
pdf.set_title('API Endpoints Documentation')
pdf.set_auto_page_break(auto=True, margin=15)
# Add a cover page
pdf.add_page()
pdf.set_font('Arial', 'B', 24)
pdf.cell(0, 20, 'API Endpoints Documentation', ln=True, align='C')
# Iterate over the JSON data and add repository name and files with code to the PDF
pdf.set_font('Arial', 'B', 16)
for repository, files in data.items():
pdf.add_page()
```

```
pdf.set_font('Arial', 'B', 14)
# Handle Unicode characters in repository name
repository = repository.encode('latin-1', 'replace').decode('latin-1')
pdf.cell(0, 10, f'Repository: {repository}', ln=True, align='L')
pdf.set_font('Arial', '', 12)
for filename, content in files.items():
pdf.multi_cell(10, 10, '')
pdf.multi_cell(0, 10, f'File: {filename}', 'B')
pdf.set_font('Courier', '', 10)
# Handle Unicode characters in code content
code = content['code'].encode('latin-1', 'replace').decode('latin-1')
code_lines = code.split('\n')
for line in code_lines:
line = line.strip()  # Remove unnecessary spaces from the beginning and end of the line
pdf.multi_cell(0, 5, line)  # Adjust the line spacing to make it more compact
# Save the PDF document with UTF-8 encoding
pdf.output(f'pdf_data/api_endpoints_{username}.pdf', 'F')
```

File: embeddings.py

```
import os
import time
import pinecone
from dotenv import load_dotenv
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.llms import OpenAI
from langchain.vectorstores import Chroma
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.chains import RetrievalQA
from langchain.vectorstores import Chroma, Pinecone
from langchain.chains.question_answering import load_qa_chain

# Load environment variables from .env file
load_dotenv()
api_key = os.environ['OPEN_AI_KEY']
pinecone_key = os.environ['PINECONE_API_KEY']
pinecone_env = os.environ['PINECONE_API_ENV']


def embeddings(username, service):
responses=[]
if(service=="chroma_service"):
responses=chroma_embedding(username)
elif(service=="pinecone_service"):
responses=pinecone_embedding(username)
else:
pass
return responses
```

```
def chroma_embedding(username):
start_time = time.time()
OPENAI_API_KEY = os.environ.get(
'OPENAI_API_KEY', api_key)
embeddings = OpenAIEmbeddings(openai_api_key=OPENAI_API_KEY)
loader = PyPDFLoader(f"pdf_data/api_endpoints_{username}.pdf")
data = loader.load()
text_splitter = RecursiveCharacterTextSplitter(
chunk_size=2000, chunk_overlap=0)
texts = text_splitter.split_documents(data)
print(f'Now you have {len(texts)} documents')
# print(texts[:100])
print("Upserting the documents.....................................")
docsearch = Chroma.from_documents(texts, embeddings)
chain = RetrievalQA.from_chain_type(
llm=OpenAI(temperature=0, openai_api_key=OPENAI_API_KEY,
max_tokens=150),
chain_type="stuff",
retriever=docsearch.as_retriever()
)
responses = []
print("Trying to load the query.....................................")
# query = "The document contains the repository along with the codes. Now you need to
decide which Repository contains the most complex code. Tell the name of the repository.
Also explain in 100 words why you think the repository is the most complex"
query1 = '''Your task is to identify the repository that contains the most complex code.
Provide your response in the following format:

'Name: [Name of the repository]'

For instance:

'Name: Blockdemon'

Please analyze the document and determine the repository with the most complex code.'''

# Run the first query.
response1 = chain.run(query1)
# Pass the previous response to the second prompt through string formatting for better
response and avoiding any conflicts between the answers.
query2 = f'''Why do you think the repository {response1[7:]} is complex'''

# Run the second query
response2 = chain.run(query2)
end_time = time.time()
print(response2)
print(response1)
responses.append(response1)
responses.append(response2)
responses.append(end_time-start_time)
return responses


def pinecone_embedding(username):
```

```
# Start the timer
start_time = time.time()
OPENAI_API_KEY = os.environ.get('OPENAI_API_KEY', api_key)
PINECONE_API_KEY = os.environ.get('PINECONE_API_KEY', pinecone_key)
PINECONE_API_ENV = os.environ.get('PINECONE_API_ENV', pinecone_env)
embeddings = OpenAIEmbeddings(openai_api_key=OPENAI_API_KEY)
# initialize pinecone
pinecone.init(
api_key=PINECONE_API_KEY,  # find at app.pinecone.io
environment=PINECONE_API_ENV  # next to api key in console
)
index_name = "testing"
print("Deleting the previous vectors.....................")
index = pinecone.Index(index_name)
index.delete(deleteAll=True)
# put in the name of your pinecone index here

loader = PyPDFLoader(f"pdf_data/api_endpoints_{username}.pdf")
data = loader.load()


text_splitter = RecursiveCharacterTextSplitter(chunk_size=2000, chunk_overlap=0)
texts = text_splitter.split_documents(data)

print (f'Now you have {len(texts)} documents')
# print(texts[:100])

print("Upserting the documents.....................................")
docsearch = Pinecone.from_texts([t.page_content for t in texts], embeddings,
index_name=index_name)
llm = OpenAI(temperature=0, openai_api_key=OPENAI_API_KEY, max_tokens=150)
chain = load_qa_chain(llm, chain_type="stuff")
responses = []
print("Trying to load the query.....................................")
# query = "The document contains the repository along with the codes. Now you need to
decide which Repository contains the most complex code. Tell the name of the repository.
Also explain in 100 words why you think the repository is the most complex"
query1 = '''Your task is to identify the repository that contains the most complex code.
Provide your response in the following format:

'Name: [Name of the repository]'

For instance:

'Name: Blockdemon'

Please analyze the document and determine the repository with the most complex code.'''

# Run the first query.
# response1 = chain.run(query1)
docs = docsearch.similarity_search(query1)
response1=chain.run(input_documents=docs, question=query1)
print(response1)
# Pass the previous response to the second prompt through string formatting for better
```

```
response and avoiding any conflicts between the answers.
query2 = f'''Desribe the repository {response1[7:]} in few lines'''

# Run the second query
# response2 = chain.run(query2)
docs = docsearch.similarity_search(query2)
response2=chain.run(input_documents=docs, question=query2)
print(response2)
print(response1)
responses.append(response1)
responses.append(response2)
index.delete(deleteAll=True)
end_time = time.time()
responses.append(end_time-start_time)
return responses
```

File: main.py

```
import uvicorn
from fastapi import FastAPI
from fastapi import FastAPI
from pydantic import BaseModel
from starlette.responses import Response
from fastapi.middleware.cors import CORSMiddleware
from data import fetch_github, convert_to_pdf
from embeddings import embeddings
from msgsend import send_email

# Initialize the fast API
app = FastAPI()


headers = {
'Accept-Language': 'en-US,en;q=0.9',
'Connection': 'keep-alive',
'Referer': 'https://trackapi.nutritionix.com/docs/',
'Sec-Fetch-Dest': 'empty',
'Sec-Fetch-Mode': 'cors',
'Sec-Fetch-Site': 'same-origin',
'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/114.0.0.0 Safari/537.36',
'accept': 'application/json',
'sec-ch-ua': '"Not.A/Brand";v="8", "Chromium";v="114", "Google Chrome";v="114"',
'sec-ch-ua-mobile': '?0',
'sec-ch-ua-platform': '"Windows"',
}

# Added multiple origins to remove the cors errors which we may encounter later

# origins = [
#     "http://localhost",
#     "http://127.0.0.1",
```

```
#     "http://127.0.0.1:8000",
#     "http://localhost:8000",
#     "http://localhost:3000",
#     "http://localhost:8000/api",
#     "https://frontend-repo-uwdc-2x0unaogp-asifrahaman13.vercel.app/"
#     "http://localhost:3000/frontend-repo"
#     "https://frontend-repo-uwdc-2x0unaogp-asifrahaman13.vercel.app/front-end"
# ]

origins = [
"http://localhost",
"http://127.0.0.1",
"http://127.0.0.1:8000",
"http://localhost:8000",
"http://localhost:3000",
"http://localhost:8000/api",
"http://192.168.140.47:3001/frontend-repo",
"http://192.168.140.47:3001/frontend-repo/",
"http://192.168.140.47:3001",
"http://192.168.140.47:3001/",
"https://frontend-repo-uwdc-2x0unaogp-asifrahaman13.vercel.app/",
"http://localhost:3000/frontend-repo",
"http://localhost:3000/frontend-repo/",
"https://64a1395b4d75163333233c5c--cheerful-sprinkles-c3bc19.netlify.app/",
"https://64a1395b4d75163333233c5c--cheerful-sprinkles-c3bc19.netlify.app",
"https://frontend-repo-uwdc-2x0unaogp-asifrahaman13.vercel.app/front-end"
"https://frontend-repo-uwdc-2x0unaogp-asifrahaman13.vercel.app/front-end/"
]



# Middleware to pass on the cors error and to check the credentials
app.add_middleware(
CORSMiddleware,
allow_origins=origins,
allow_credentials=True,
allow_methods=["*"],
allow_headers=["*"],
)

# Create the data model to define the data types of the json data we will accept.
class YourDataModel(BaseModel):
username: str
email_id: str
service: str

# Function to fetch all the codes of the repositories of the person

def fetching(username,email_id,service):
fetching_info=fetch_github(username)
convert_to_pdf(username)
responses = embeddings(username, service)
if(len(email_id))!=0:
send_email(username,email_id, responses,fetching_info, responses)
```

```
return responses

@app.post("/postdata")
async def your_endpoint(your_data: YourDataModel):
# Access the JSON data within the endpoint
username = your_data.username
email_id = your_data.email_id
service = your_data.service
# Process the data as needed
# Example: Return a response message with the received data
msg = fetching(username,email_id,service)
return {"response1": f"{msg[0]}", "response2": f"{msg[1]}"}


# API to access the index page of the web application
@app.get("/hello")
async def index():
response = Response(content="Hello, World!", media_type="text/plain")
return response

# Driving code of the file.
if __name__ == "__main__":
uvicorn.run(app, host="0.0.0.0", port=8000)


# Lalit2005
# VanRoy
# priyanshu9588
# abir-taheer
```

File: msgsend.py

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.application import MIMEApplication
import datetime
from data import convert_to_formatted_pdf

def send_email(username,receiver_email, responses, fetching_info, embedding_info,
sender_email='asifrahaman162@gmail.com', sender_password='lcpfzddrihzynspf'):
body = "<html><body>"
body += "<h1><p>Here is the feedback of the repository:</p></h1>"
name = responses[0]
reasons = responses[1]
body += f"<p><b>Name of the most complex repository:</b> {name}</p>"
body += f"<p><b>Reasons (BY AI):</b></p>"
body += f"{reasons}"
body += "<p><b>Here are some user details:</b></p>"
if(fetching_info[0] != None and len(fetching_info[0]) > 0):
body += f"<b>Name:</b>"
body += f"<p>{fetching_info[0]}</p>"
if(fetching_info[1] != None and len(fetching_info[1]) > 0):
```

```
body += f"<b>Username:</b>"
body += f"<p>{fetching_info[1]}</p>"
if(fetching_info[2] != None and len(fetching_info[2]) > 0):
body += f"<b>Location:</b>"
body += f"<p>{fetching_info[2]}</p>"
if(fetching_info[3] != None and len(fetching_info[3]) > 0):
body += f"<b>Bio:</b>"
body += f"<p>{fetching_info[3]}</p>"
body += f"<p><b>Total time taken to fetch all the codes:</b> {int(fetching_info[-1])}
seconds / {int((fetching_info[-1])/60)} min</p>"
body += f"<p><b>Total time taken to decide the most complex repository by AI:</b>
{int(embedding_info[-1])} seconds/ {int((embedding_info[-1])/60)} min</p>"
body += "<br><br>"
body += "<p>Thanks,</p>"
body += "<p>Team GitCrawler</p>"
body += "</body></html>"

# Set up the SMTP server
# Change this if you're using a different email provider
smtp_server = "smtp.gmail.com"
smtp_port = 587  # Change this if required
smtp_username = sender_email
subject = "Github Repository Feedback"

# Create a multipart message and set the headers
message = MIMEMultipart()
# Attach the PDF file
convert_to_formatted_pdf(username)
with open(f"pdf_data/api_endpoints_{username}.pdf", "rb") as file:
attachment = MIMEApplication(file.read(), _subtype="pdf")
attachment.add_header(
"Content-Disposition", "attachment", filename=(f"{username}_repositories.pdf")
)
message.attach(attachment)
message["From"] = sender_email
message["To"] = receiver_email
message["Subject"] = subject + " " + \
str(datetime.date.today()) + " " + \
str(datetime.datetime.now().strftime("%H:%M"))

# Add body to the email as HTML content
message.attach(MIMEText(body, "html"))

try:
# Login to the SMTP server
server = smtplib.SMTP(smtp_server, smtp_port)
server.starttls()
server.login(smtp_username, sender_password)

# Send the email
server.sendmail(sender_email, receiver_email, message.as_string())

# Cleanup
server.quit()
```

```
print(f"Email sent successfully to {receiver_email}")
except Exception as e:
print("An error occurred while sending the email:", str(e))


# send_email("asifrahaman13","asifr@iitbhilai.ac.in", ["neon", "There are lots of reason as to why this is the most comples.", 45.14], [
#                         "nil",  "Hello  I  am  nil",  "Australia","sdfsdf"  ,41],  [14], sender_email='asifrahaman162@gmail.com', sender_password='lcpfzddrihzynspf')
```

Repository: backendportfolio

File: connection.js

```
require('dotenv').config();

const mongoose=require('mongoose');

mongoose.connect(process.env.CONNECTION_URL,{ useNewUrlParser: true, useUnifiedTopology:
true },async()=>{
console.log("Connected successfully to mongoose.")
});
```

File: index.js

```
require('dotenv').config();
require('./db/connection')
var cors=require('cors')
const Contact=require('./models/contacts')
const Student=require('./models/students')
const Skills=require('./models/skills')
const Participation=require('./models/participations')
const Hackathon=require('./models/hackathons')
const express = require("express");
const app = express();

const PORT=8000;

const port=process.env.PORT || PORT;

app.use(cors());
app.use(express.json());

app.get("/",async(req,res)=>{
const user= await Student.find();
res.send(user)
})

app.get("/participations",async(req,res)=>{
const participation= await Participation.find();
res.send(participation)
})

app.post("/participations",async(req,res)=>{
const                                                                    result=await
Participation({participation:req.body.participation,name:req.body.participation})
await result.save();
res.send(result)
})
```

```
app.post("/contacts",async (req,res)=>{
const result=await Contact({name:req.body.name,email:req.body.email});
await result.save();
res.send(result)
})


app.get("/skills",async (req,res)=>{
const result=await Skills.find();
res.send(result)
})


app.get("/hackathon",async (req,res)=>{
const result=await Hackathon.find();
res.send(result)
})

app.listen(port,()=>{
console.log(`Listening on port: ${PORT}`)
})
```

File: contacts.js

```
const mongoose = require('mongoose')

const ContactSchema=new mongoose.Schema({
name:{
type:String
},
email:{
type:String
},
profession:{
type:String
},
age:{
type:Number
},
concern:{
type:String
}
})

const contact=mongoose.model('contact',ContactSchema)

module.exports =contact
```

File: hackathons.js

```
const mongoose = require('mongoose')

const HackathonSchema=new mongoose.Schema({
title:{
type:String
},
name:{
type:String
},
image:{
type:String
}
})

const Hackathon=mongoose.model('hackathon', HackathonSchema)
module.exports= Hackathon
```

File: participations.js

```
const mongoose=require('mongoose');

const ParticipationSchema=new mongoose.Schema({
participation:{
type:String
},
name:{
type:String
}
})

const participation=mongoose.model('participation',ParticipationSchema)
module.exports=participation
```

File: skills.js

```
const mongoose = require('mongoose');

const SkillsSchema=new mongoose.Schema({
skill:{
type:String
},
proficiency:{
type:String
}
})

const skills=mongoose.model('Skills',SkillsSchema)
module.exports =skills
```

File: students.js

```
const mongoose=require('mongoose');

const StudentSchema=new mongoose.Schema({
name:{
type:String
},
email:{
type:String
},
phone:{
type:Number
}
})

const Student=new mongoose.model("studentcollection",StudentSchema);

module.exports = Student
```

Repository: blog-api-endpoints

File: blogTest.test.js

```
const request = require("supertest");
const baseURL = "http://localhost:5000";

// Given user already exist in the data base

let authToken = "";
let success = false;
let blogId = "";

beforeAll(async () => {
const response = await request(baseURL).post("/api/auth/login").send({
email: "rex@test.com",
password: "@Atest98",
});
authToken = response.body.authtoken;
success = response.body.success;
});

afterAll(async () => {
// const response = await request(baseURL).post("/api/auth/login").send({
//     email:"rex@test.com",
//     password:"@Atest98",

//   })
authToken = ""; //reset access token
});

describe("POST /addblog", () => {
it("should return 200", async () => {
const response = await request(baseURL)
.post("/api/blogs/addblog")
.set("auth-token", authToken)

.send({
title: "Dummy title 2",
description: "dummy description 2",
});
blogId = response.body._id;

expect(response.statusCode).toBe(200);
expect(response.body.createdby).toBe("General");
});
});

describe("POST /updateblog", () => {
it("should return 200", async () => {
const response = await request(baseURL)
.put(`/api/blogs/updateblog/${blogId}`)
```

```
.set("auth-token", authToken)
.send({
title: "update Dummy title 2",
description: "dummy description 2",
});

expect(response.statusCode).toBe(200);
expect(response.body.blog.title).toBe("update Dummy title 2");
});
});

describe("POST /deleteblog", () => {
it("should return 200", async () => {
const response = await request(baseURL)
.delete(`/api/blogs/deleteblog/${blogId}`)
.set("auth-token", authToken);

expect(response.statusCode).toBe(200);
expect(response.body.Success).toBe("Blog has been deleted");
});
});
```

File: db.js

```
const mongoose = require('mongoose');

// Mongodb atlas connection URL
const                                    mongoURI                                    =
"mongodb+srv://asifr:asifrahaman@cluster0.nelr8ne.mongodb.net/?retryWrites=true&w=majori
ty"

// Function to connect to mongodb atlas
const connectToMongo =  ()=>{
mongoose.connect(mongoURI, ()=>{
console.log("Connected to Mongo Successfully");
})
}

module.exports = connectToMongo;
```

File: index.js

```
const connectToMongo = require('./db');
const express = require('express')
var cors = require('cors')

// Connect to MongoDB database
connectToMongo();
// Use express module in Node JS
const app = express()
```

```
// Define the port where you need to run the server
const port = 5000

// Use cors to avoid cors error
app.use(cors())

// Use json to let the application know how to use json object
app.use(express.json())

// Available Routes
app.use('/api/auth', require('./routes/auth'))
app.use('/api/blogs', require('./routes/blogs'))

// Listen to the ports
app.listen(port, () => {
console.log(`iblogbook backend listening at http://localhost:${port}`)
})

module.exports=app
```

File: fetchuser.js

```
var jwt = require('jsonwebtoken');
const JWT_SECRET = 'Hello world';

const fetchuser = (req, res, next) => {
// Get the user from the jwt token and add id to req object
const token = req.header('auth-token');
if (!token) {
res.status(401).send({ error: "Please authenticate using a valid token" })
}
try {
// Verify the token
const data = jwt.verify(token, JWT_SECRET);
req.user = data.user;
// If token is valid then let the user to make further requests
next();
} catch (error) {
res.status(401).send({ error: "Please authenticate using a valid token" })
}

}


module.exports = fetchuser;
```

File: Blog.js

```
const mongoose = require('mongoose');
const { Schema } = mongoose;

// Make a schema object for the blog
```

```
const BlogSchema = new Schema({
user:{
type: mongoose.Schema.Types.ObjectId,
ref: 'user'
},
title:{
type: String,
required: true
},
description:{
type: String,
required: true,
},
createdby:{
type: String,
default: "General"
},
date:{
type: Date,
default: Date.now
},
});

// Create a blogs model
module.exports = mongoose.model('blogs', BlogSchema);
```

File: User.js

```
const mongoose = require("mongoose");
const { Schema } = mongoose;

// Create a user schema
const UserSchema = new Schema({
name: {
type: String,
required: true,
},
email: {
type: String,
required: true,
unique: true,
},
password: {
type: String,
required: true,
},
date: {
type: Date,
default: Date.now,
},
});

// Create a user schema
```

```
const User = mongoose.model("user", UserSchema);
module.exports = User;
```

File: auth.js

```
const express = require('express');
const User = require('../models/User');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const bcrypt = require('bcryptjs');
var jwt = require('jsonwebtoken');
// Use middleware for authentication purpose
var fetchuser = require('../middleware/fetchuser');

// Use any text for secret key
const JWT_SECRET = 'Hello world';

// ROUTE 1: Create a User using: POST "/api/auth/createuser". No login required
router.post('/createuser', [
body('name', 'Enter a valid name').isLength({ min: 3 }),
body('email', 'Enter a valid email').isEmail(),
body('password', 'Password must be atleast 5 characters').isLength({ min: 5 }),
], async (req, res) => {
// If there are errors, return Bad request and the errors
const errors = validationResult(req);
if (!errors.isEmpty()) {
return res.status(400).json({ errors: errors.array() });
}
try {
// Check whether the user with this email exists already
let user = await User.findOne({ email: req.body.email });
if (user) {
return res.status(400).json({ error: "Sorry a user with this email already exists" })
}
// Add salt as per the prefernce
const salt = await bcrypt.genSalt(10);
// Hashing the password to add security
const secPass = await bcrypt.hash(req.body.password, salt);

// Create a new user
user = await User.create({
name: req.body.name,
password: secPass,
email: req.body.email,
});
// Define the data to exist in the auth token
const data = {
user: {
id: user.id
}
}
const authtoken = jwt.sign(data, JWT_SECRET);
```

```
// res.json(user)
res.json({ authtoken })

} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})


// ROUTE 2: Authenticate a User using: POST "/api/auth/login". No login required
router.post('/login', [
body('email', 'Enter a valid email').isEmail(),
body('password', 'Password cannot be blank').exists(),
], async (req, res) => {
let success = false;
// If there are errors, return Bad request and the errors
const errors = validationResult(req);
if (!errors.isEmpty()) {
return res.status(400).json({ errors: errors.array() });
}

const { email, password } = req.body;
try {
let user = await User.findOne({ email });
if (!user) {
success = false
return res.status(400).json({ error: "Please try to login with correct credentials" });
}
// Compare the password entered with the password set using bcrypt
const passwordCompare = await bcrypt.compare(password, user.password);
if (!passwordCompare) {
// If password does not match then make success to false
success = false
// Send error message
return res.status(400).json({ success, error: "Please try to login with correct
credentials" });
}
// Define the data that you want to store in auth token
const data = {
user: {
id: user.id
}
}
const authtoken = jwt.sign(data, JWT_SECRET);
// If everything is allright then make the value of success true.
success = true;
// Send a sucess message and a auth token
res.json({ success, authtoken })

} catch (error) {
// Send error message if anything is wrong
console.error(error.message);
res.status(500).send("Internal Server Error");
```

```
}


});


// ROUTE 3: Get loggedin User Details using: POST "/api/auth/getuser". Login required
router.post('/getuser', fetchuser,  async (req, res) => {

try {
userId = req.user.id;
const user = await User.findById(userId).select("-password")
res.send(user)
} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})
module.exports = router
```

File: blogs.js

```
const express = require('express');
const router = express.Router();
const fetchuser = require('../middleware/fetchuser');
const Blog = require('../models/Blog');
const { body, validationResult } = require('express-validator');

// ROUTE 1: Get All the Blogs using: GET "/api/blogs/getuser". Login required
router.get('/fetchallblogs', fetchuser, async (req, res) => {
try {
const blogs = await Blog.find({ user: req.user.id });
res.json(blogs)
} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})

// ROUTE 2: Add a new Blog using: POST "/api/blogs/addblog". Login required
router.post('/addblog', fetchuser, [
body('title', 'Enter a valid title').isLength({ min: 3 }),
body('description', 'Description must be atleast 5 characters').isLength({ min: 5 }),],
async (req, res) => {
try {
// Destructure the body to get the values of the blog parameters
const { title, description, createdby } = req.body;

// If there are errors, return Bad request and the errors
const errors = validationResult(req);
// If there is any error then return an error message
if (!errors.isEmpty()) {
return res.status(400).json({ errors: errors.array() });
```

```
}
const blog = new Blog({
title, description, createdby, user: req.user.id
})
const savedBlog = await blog.save()

res.json(savedBlog)

} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})

// ROUTE 3: Update an existing Blog using: PUT "/api/blogs/updateblog". Login required
router.put('/updateblog/:id', fetchuser, async (req, res) => {
const { title, description, createdby } = req.body;
try {
// Create a newBlog object
const newBlog = {};
if (title) { newBlog.title = title };
if (description) { newBlog.description = description };
if (createdby) { newBlog.createdby = createdby };

// Find the blog to be updated and update it
let blog = await Blog.findById(req.params.id);
if (!blog) { return res.status(404).send("Not Found") }

if (blog.user.toString() !== req.user.id) {
return res.status(401).send("Not Allowed");
}
blog = await Blog.findByIdAndUpdate(req.params.id, { $set: newBlog }, { new: true })
res.json({ blog });
} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})

// ROUTE 4: Delete an existing Blog using: DELETE "/api/blogs/deleteblog". Login
required
router.delete('/deleteblog/:id', fetchuser, async (req, res) => {
try {
// Find the blog to be delete and delete it
let blog = await Blog.findById(req.params.id);
if (!blog) { return res.status(404).send("Not Found") }

// Allow deletion only if user owns this Blog
if (blog.user.toString() !== req.user.id) {
return res.status(401).send("Not Allowed");
}

blog = await Blog.findByIdAndDelete(req.params.id)
res.json({ "Success": "Blog has been deleted", blog: blog });
```

```
} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})
module.exports = router
```

Repository: blog-endpoints

File: db.js

```
const mongoose = require('mongoose');

const mongoURI = "mongodb+srv://asifr:asifrahaman@cluster0.nelr8ne.mongodb.net/?retryWrites=true&w=majority"

const connectToMongo = ()=>{
mongoose.connect(mongoURI, ()=>{
console.log("Connected to Mongo Successfully");
})
}

module.exports = connectToMongo;
```

File: index.js

```
const connectToMongo = require('./db');
const express = require('express')
var cors = require('cors')

connectToMongo();
const app = express()
const port = 5000

app.use(cors())
app.use(express.json())

// Available Routes
app.use('/api/auth', require('./routes/auth'))
app.use('/api/blogs', require('./routes/blogs'))


app.listen(port, () => {
console.log(`iblogbook backend listening at http://localhost:${port}`)
})
```

File: fetchuser.js

```
var jwt = require('jsonwebtoken');
const JWT_SECRET = 'Hello world';

const fetchuser = (req, res, next) => {
// Get the user from the jwt token and add id to req object
const token = req.header('auth-token');
if (!token) {
res.status(401).send({ error: "Please authenticate using a valid token" })
```

```
}
try {
const data = jwt.verify(token, JWT_SECRET);
req.user = data.user;
next();
} catch (error) {
res.status(401).send({ error: "Please authenticate using a valid token" })
}

}


module.exports = fetchuser;
```

File: Blog.js

```
const mongoose = require('mongoose');
const { Schema } = mongoose;

const NotesSchema = new Schema({
user:{
type: mongoose.Schema.Types.ObjectId,
ref: 'user'
},
title:{
type: String,
required: true
},
description:{
type: String,
required: true,
},
createdby:{
type: String,
default: "General"
},
date:{
type: Date,
default: Date.now
},
});

module.exports = mongoose.model('blogs', NotesSchema);
```

File: User.js

```
const mongoose = require('mongoose');
const { Schema } = mongoose;

const UserSchema = new Schema({
name:{
type: String,
```

```
required: true
},
email:{
type: String,
required: true,
unique: true
},
password:{
type: String,
required: true
},
date:{
type: Date,
default: Date.now
},
});
const User = mongoose.model('user', UserSchema);
module.exports = User;
```

File: auth.js

```
const express = require('express');
const User = require('../models/User');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const bcrypt = require('bcryptjs');
var jwt = require('jsonwebtoken');
var fetchuser = require('../middleware/fetchuser');

const JWT_SECRET = 'Hello world';

// ROUTE 1: Create a User using: POST "/api/auth/createuser". No login required
router.post('/createuser', [
body('name', 'Enter a valid name').isLength({ min: 3 }),
body('email', 'Enter a valid email').isEmail(),
body('password', 'Password must be atleast 5 characters').isLength({ min: 5 }),
], async (req, res) => {
// If there are errors, return Bad request and the errors
const errors = validationResult(req);
if (!errors.isEmpty()) {
return res.status(400).json({ errors: errors.array() });
}
try {
// Check whether the user with this email exists already
let user = await User.findOne({ email: req.body.email });
if (user) {
return res.status(400).json({ error: "Sorry a user with this email already exists" })
}
const salt = await bcrypt.genSalt(10);
const secPass = await bcrypt.hash(req.body.password, salt);

// Create a new user
user = await User.create({
```

```
name: req.body.name,
password: secPass,
email: req.body.email,
});
const data = {
user: {
id: user.id
}
}
const authtoken = jwt.sign(data, JWT_SECRET);
// res.json(user)
res.json({ authtoken })

} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})


// ROUTE 2: Authenticate a User using: POST "/api/auth/login". No login required
router.post('/login', [
body('email', 'Enter a valid email').isEmail(),
body('password', 'Password cannot be blank').exists(),
], async (req, res) => {
let success = false;
// If there are errors, return Bad request and the errors
const errors = validationResult(req);
if (!errors.isEmpty()) {
return res.status(400).json({ errors: errors.array() });
}

const { email, password } = req.body;
try {
let user = await User.findOne({ email });
if (!user) {
success = false
return res.status(400).json({ error: "Please try to login with correct credentials" });
}

const passwordCompare = await bcrypt.compare(password, user.password);
if (!passwordCompare) {
success = false
return  res.status(400).json({  success,  error:  "Please  try  to  login  with  correct
credentials" });
}

const data = {
user: {
id: user.id
}
}
const authtoken = jwt.sign(data, JWT_SECRET);
success = true;
```

```
res.json({ success, authtoken })

} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}


});


// ROUTE 3: Get loggedin User Details using: POST "/api/auth/getuser". Login required
router.post('/getuser', fetchuser,  async (req, res) => {

try {
userId = req.user.id;
const user = await User.findById(userId).select("-password")
res.send(user)
} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})
module.exports = router
```

File: blogs.js

```
const express = require('express');
const router = express.Router();
const fetchuser = require('../middleware/fetchuser');
const Note = require('../models/Blog');
const { body, validationResult } = require('express-validator');

// ROUTE 1: Get All the Notes using: GET "/api/blogs/getuser". Login required
router.get('/fetchallblogs', fetchuser, async (req, res) => {
try {
const blogs = await Note.find({ user: req.user.id });
res.json(blogs)
} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})

// ROUTE 2: Add a new Note using: POST "/api/blogs/addblog". Login required
router.post('/addblog', fetchuser, [
body('title', 'Enter a valid title').isLength({ min: 3 }),
body('description', 'Description must be atleast 5 characters').isLength({ min: 5 }),],
async (req, res) => {
try {
const { title, description, createdby } = req.body;

// If there are errors, return Bad request and the errors
```

```
const errors = validationResult(req);
if (!errors.isEmpty()) {
return res.status(400).json({ errors: errors.array() });
}
const blog = new Note({
title, description, createdby, user: req.user.id
})
const savedNote = await blog.save()

res.json(savedNote)

} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})

// ROUTE 3: Update an existing Note using: PUT "/api/blogs/updateblog". Login required
router.put('/updateblog/:id', fetchuser, async (req, res) => {
const { title, description, createdby } = req.body;
try {
// Create a newNote object
const newNote = {};
if (title) { newNote.title = title };
if (description) { newNote.description = description };
if (createdby) { newNote.createdby = createdby };

// Find the blog to be updated and update it
let blog = await Note.findById(req.params.id);
if (!blog) { return res.status(404).send("Not Found") }

if (blog.user.toString() !== req.user.id) {
return res.status(401).send("Not Allowed");
}
blog = await Note.findByIdAndUpdate(req.params.id, { $set: newNote }, { new: true })
res.json({ blog });
} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})

// ROUTE 4: Delete an existing Note using: DELETE "/api/blogs/deleteblog". Login
required
router.delete('/deleteblog/:id', fetchuser, async (req, res) => {
try {
// Find the blog to be delete and delete it
let blog = await Note.findById(req.params.id);
if (!blog) { return res.status(404).send("Not Found") }

// Allow deletion only if user owns this Note
if (blog.user.toString() !== req.user.id) {
return res.status(401).send("Not Allowed");
}
```

```
blog = await Note.findByIdAndDelete(req.params.id)
res.json({ "Success": "Note has been deleted", blog: blog });
} catch (error) {
console.error(error.message);
res.status(500).send("Internal Server Error");
}
})
module.exports = router
```

Repository: Blogging-website

File: __init__.py

File: admin.py

```
from django.contrib import admin
from .models import Contact

# Register your models here.
admin.site.register(Contact)
```

File: apps.py

```
from django.apps import AppConfig


class HomeConfig(AppConfig):
default_auto_field = 'django.db.models.BigAutoField'
name = 'home'
```

File: 0001_initial.py

```
# Generated by Django 4.0.1 on 2022-03-04 19:02

from django.db import migrations, models


class Migration(migrations.Migration):

initial = True

dependencies = [
]

operations = [
migrations.CreateModel(
name='Contact',
fields=[
('sno', models.AutoField(primary_key=True, serialize=False)),
('name', models.CharField(max_length=255)),
('phone', models.CharField(max_length=13)),
('email', models.CharField(max_length=100)),
('content', models.TextField()),
('timeStamp', models.DateTimeField(auto_now_add=True)),
```

```
],
),
]
```

File: 0002_alter_post_content.py

```
# Generated by Django 4.0.1 on 2022-03-05 03:08

from django.db import migrations, models


class Migration(migrations.Migration):

dependencies = [
('blog', '0001_initial'),
]

operations = [
migrations.AlterField(
model_name='post',
name='content',
field=models.CharField(max_length=5000),
),
]
```

File: 0003_post_slug.py

```
# Generated by Django 4.0.3 on 2022-03-05 03:13

from django.db import migrations, models


class Migration(migrations.Migration):

dependencies = [
('blog', '0002_alter_post_content'),
]

operations = [
migrations.AddField(
model_name='post',
name='slug',
field=models.CharField(default='', max_length=100),
),
]
```

File: 0004_remove_post_timestamp.py

```
# Generated by Django 4.0.1 on 2022-03-05 09:47

from django.db import migrations


class Migration(migrations.Migration):

dependencies = [
('blog', '0003_post_slug'),
]

operations = [
migrations.RemoveField(
model_name='post',
name='timeStamp',
),
]
```

File: 0005_post_timestamp.py

```
# Generated by Django 4.0.1 on 2022-03-05 09:49

from django.db import migrations, models
import django.utils.timezone


class Migration(migrations.Migration):

dependencies = [
('blog', '0004_remove_post_timestamp'),
]

operations = [
migrations.AddField(
model_name='post',
name='timeStamp',
field=models.DateTimeField(auto_now_add=True, default=django.utils.timezone.now),
preserve_default=False,
),
]
```

File: 0006_alter_post_timestamp.py

```
# Generated by Django 4.0.1 on 2022-03-06 13:43

from django.db import migrations, models


class Migration(migrations.Migration):
```

```
dependencies = [
('blog', '0005_post_timestamp'),
]

operations = [
migrations.AlterField(
model_name='post',
name='timeStamp',
field=models.DateField(),
),
]
```

File: 0007_alter_post_timestamp.py

```
# Generated by Django 4.0.1 on 2022-03-06 13:45

from django.db import migrations, models


class Migration(migrations.Migration):

dependencies = [
('blog', '0006_alter_post_timestamp'),
]

operations = [
migrations.AlterField(
model_name='post',
name='timeStamp',
field=models.DateTimeField(auto_now_add=True),
),
]
```

File: 0008_alter_post_author_alter_post_content_alter_post_slug_and_more.py

```
# Generated by Django 4.0.1 on 2022-03-06 13:51

from django.db import migrations, models


class Migration(migrations.Migration):

dependencies = [
('blog', '0007_alter_post_timestamp'),
]

operations = [
migrations.AlterField(
model_name='post',
name='author',
```

```
field=models.CharField(max_length=14),
),
migrations.AlterField(
model_name='post',
name='content',
field=models.TextField(),
),
migrations.AlterField(
model_name='post',
name='slug',
field=models.CharField(max_length=130),
),
migrations.AlterField(
model_name='post',
name='timeStamp',
field=models.DateTimeField(blank=True),
),
]
```

File: models.py

```
from django.db import models
class Contact(models.Model):
sno= models.AutoField(primary_key=True)
name= models.CharField(max_length=255)
phone= models.CharField(max_length=13)
email= models.CharField(max_length=100)
content= models.TextField()
timeStamp=models.DateTimeField(auto_now_add=True, blank=True)

def __str__(self):
return "Message from " + self.name + ' - ' + self.email
```

File: tests.py

```
from django.test import TestCase

# Create your tests here.
```

File: urls.py

```
"""icoder URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
https://docs.djangoproject.com/en/4.0/topics/http/urls/
Examples:
Function views
1. Add an import:  from my_app import views
```

```
2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
1. Add an import:  from other_app.views import Home
2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include

admin.site.site_header="Icon admins(Asif Rahaman)"
admin.site.site_title="Icon admins"
admin.site.site_title="Icon admins(Asif Rahaman)"

urlpatterns = [
path('admin/', admin.site.urls),
path('', include('home.urls')),
path('blog/', include('blog.urls')),
]
# If the path is included with the blog then the include will handle the rest of the
work
```

File: asgi.py

```
"""
ASGI config for icoder project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/4.0/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'icoder.settings')

application = get_asgi_application()
```

File: settings.py

```
"""
Django settings for icoder project.

Generated by 'django-admin startproject' using Django 4.0.1.

For more information on this file, see
https://docs.djangoproject.com/en/4.0/topics/settings/
```

```
For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.0/ref/settings/
"""

import os
from pathlib import Path
from django.contrib import messages
# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-yfg%db=pyy+ida3qp7dit!^t!g3!2%flg$@mcf4whubp=#nhoi'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'home.apps.HomeConfig',
'blog.apps.BlogConfig',
'django.contrib.humanize',
# 'django.contrib.extras'
]

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'icoder.urls'

TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
```

```
'DIRS': ['templates'],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]

WSGI_APPLICATION = 'icoder.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {
'default': {
'ENGINE': 'django.db.backends.sqlite3',
'NAME': BASE_DIR / 'db.sqlite3',
}
}


# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
{
'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]


# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True
```

```
USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

STATICFILES_DIRS = [
os.path.join(BASE_DIR,  "static"),
]


DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

MESSAGE_TAGS = {
messages.ERROR:'danger'
}
```

File: views.py

```
from django.shortcuts import render, HttpResponse

# Create your views here.
def home(request):
return HttpResponse('This is home.')

def about(request):
return HttpResponse('This is about.')

def contact(request):
return HttpResponse('This is contact.')
```

File: wsgi.py

```
"""
WSGI config for icoder project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/4.0/howto/deployment/wsgi/
"""

import os
```

```
from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'icoder.settings')

application = get_wsgi_application()
```

File: manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
"""Run administrative tasks."""
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'icoder.settings')
try:
from django.core.management import execute_from_command_line
except ImportError as exc:
raise ImportError(
"Couldn't import Django. Are you sure it's installed and "
"available on your PYTHONPATH environment variable? Did you "
"forget to activate a virtual environment?"
) from exc
execute_from_command_line(sys.argv)


if __name__ == '__main__':
main()
```

File: requirements.txt

```
django==4.0.1
```

Repository: bomb-dashboard-asif-rahaman

File: bug_report.md

```
---
name: Bug report
about: Create a report to help us improve
title: ''
labels: ''
assignees: ''
---

**Describe the bug**

A clear and concise description of what the bug is.

**To Reproduce**

Steps to reproduce the behavior:

1. Go to '...'
2. Click on '....'
3. Scroll down to '....'
4. See error

**Expected behavior**

A clear and concise description of what you expected to happen.

**Screenshots**

If applicable, add screenshots to help explain your problem.

**Desktop (please complete the following information):**

- OS: [e.g. iOS]
- Browser [e.g. chrome, safari]
- Version [e.g. 22]

**Smartphone (please complete the following information):**

- Device: [e.g. iPhone6]
- OS: [e.g. iOS8.1]
- Browser [e.g. stock browser, safari]
- Version [e.g. 22]

**Additional context**

Add any other context about the problem here.
```

File: custom.md

```
---
name: Custom issue template
about: Describe this issue template's purpose here.
title: ''
labels: ''
assignees: ''
---
```

File: feature_request.md

```
---
name: Feature request
about: Suggest an idea for this project
title: ''
labels: ''
assignees: ''
---

**Is your feature request related to a problem? Please describe.**

A clear and concise description of what the problem is. Ex. I'm always frustrated when [...]

**Describe the solution you'd like**

A clear and concise description of what you want to happen.

**Describe alternatives you've considered**

A clear and concise description of any alternative solutions or features you've considered.

**Additional context**

Add any other context or screenshots about the feature request here.
```

File: PULL_REQUEST_TEMPLATE.md

File: CODE_OF_CONDUCT.md

```
# Contributor Code of Conduct

As contributors and maintainers of this project, and in the interest of fostering an open
and welcoming community, we pledge to respect all people who contribute through
```

reporting
issues, posting feature requests, updating documentation, submitting pull requests or patches, and other activities.

We are committed to making participation in this project a harassment-free experience for
everyone, regardless of level of experience, gender, gender identity and expression, sexual orientation, disability, personal appearance, body size, race, ethnicity, age, religion, or nationality.

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery
- Personal attacks
- Trolling or insulting/derogatory comments
- Public or private harassment
- Publishing other's private information, such as physical or electronic addresses, without explicit permission
- Other unethical or unprofessional conduct

Project maintainers have the right and responsibility to remove, edit, or reject comments,
commits, code, issues, and other contributions that are not aligned to this
Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors
that they deem inappropriate, threatening, offensive, or harmful.

By adopting this Code of Conduct, project maintainers commit themselves to fairly and consistently applying these principles to every aspect of managing this project. Project maintainers who do not follow or enforce the Code of Conduct may be permanently removed from the project team.

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community.

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the team in our [discord server](https://discord.gg/tombfinance) . All complaints will
be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. Maintainers are obligated to maintain confidentiality with regard to the reporter of an incident.

Repository: branding-front-end

File: README.md

```
# Getting Started with Create React App

This project was bootstrapped with [Create React App](https://github.com/facebook/create-react-app).

## Available Scripts

In the project directory, you can run:

### `yarn start`

Runs the app in the development mode.\
Open [http://localhost:3000](http://localhost:3000) to view it in your browser.

The page will reload when you make changes.\
You may also see any lint errors in the console.

### `yarn test`

Launches the test runner in the interactive watch mode.\
See the section about [running tests](https://facebook.github.io/create-react-app/docs/running-tests) for more information.

### `yarn build`

Builds the app for production to the `build` folder.\
It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.\
Your app is ready to be deployed!

See the section about [deployment](https://facebook.github.io/create-react-app/docs/deployment) for more information.

### `yarn eject`

**Note: this is a one-way operation. Once you `eject`, you can't go back!**

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.
```

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

Learn More

You can learn more in the [Create React App documentation](https://facebook.github.io/create-react-app/docs/getting-started).

To learn React, check out the [React documentation](https://reactjs.org/).

Code Splitting

This section has moved here: https://facebook.github.io/create-react-app/docs/code-splitting

Analyzing the Bundle Size

This section has moved here: https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size

Making a Progressive Web App

This section has moved here: https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app

Advanced Configuration

This section has moved here: https://facebook.github.io/create-react-app/docs/advanced-configuration

Deployment

This section has moved here: https://facebook.github.io/create-react-app/docs/deployment

`yarn build` fails to minify

This section has moved here: https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify

File: index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta name="theme-color" content="#000000" />
<meta
name="description"
content="Web site created using create-react-app"
/>
<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
<!--
manifest.json provides metadata used when your web app is installed on a
user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
-->
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<!--
Notice the use of %PUBLIC_URL% in the tags above.
It will be replaced with the URL of the `public` folder during the build.
Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
work correctly both with client-side routing and a non-root public URL.
Learn how to configure a non-root public URL by running `npm run build`.
-->
<title>React App</title>

<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
<link
href="https://fonts.googleapis.com/css2?family=Rowdies:wght@300&family=Rubik&display=swa
p"
rel="stylesheet"
/>
<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
<link
href="https://fonts.googleapis.com/css2?family=Nunito+Sans:wght@300&family=Rowdies:wght@
300&family=Rubik&display=swap"
rel="stylesheet"
/>
</head>
<body>
<noscript>You need to enable JavaScript to run this app.</noscript>
<div id="root"></div>
<!--
This HTML file is a template.
If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.
The build step will place the bundled scripts into the <body> tag.
```

```
To begin the development, run `npm start` or `yarn start`.
To create a production bundle, use `npm run build` or `yarn build`.
-->
</body>
</html>
```

File: robots.txt

```
# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
```

File: App.css

```
* {
margin: 0;
padding: 0;
}

html,body {
overflow-x: hidden;
}
```

File: App.js

```
import logo from './logo.svg';
import './App.css';
import Home from './Components/Home';

function App() {
return (
<>
<Home/>
</>
);
}

export default App;
```

File: App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
render(<App />);
const linkElement = screen.getByText(/learn react/i);
```

```
expect(linkElement).toBeInTheDocument();
});
```

File: Brands.jsx

```
import React from "react";

const Brands = () => {
return (
<>
<div className="brands">
<div className="brands-container">
<div className="brand-headers">
<div className="brand-heading">We design brands</div>
<div className="brand-header-content">
Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia
veniam quasi molestias nesciunt consequatur necessitatibus eveniet
sapiente soluta quaerat ipsa.
</div>
</div>
<div className="brand-body">
<li>lesarenics</li>
<li>avesamples</li>
<li>WHITESPACE</li>
<li>monolasm</li>
<li>barcelona</li>
<li>smoothVanilla</li>
</div>
</div>
</div>
</>
);
};

export default Brands;
```

File: Experience.jsx

```
import React from "react";

const Experience = () => {
return (
<>
<div className="experience">
<div className="experience-child">
<div className="experience-header">
<h1>We build experience</h1>
<p>
Lorem ipsum dolor sit amet consectetur adipisicing elit. In, quas!
A adipisci in itaque voluptatibus!
```

```
</p>
</div>
<div className="experience-body">
<div className="experience-cards" id="design">
<div className="experience-card-header">DESIGN</div>
<p>
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Labore
commodi error sed, odit facere quibusdam exercitationem atque
assumenda nihil laudantium ipsum enim voluptatem. Reiciendis,
mollitia?
</p>
<button className="learn-more">LEARN MORE</button>
</div>
<div className="experience-cards" id="content">
<div className="experience-card-header">CONTENT</div>
<p>
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Labore
commodi error sed, odit facere quibusdam exercitationem atque
assumenda nihil laudantium ipsum enim voluptatem. Reiciendis,
mollitia?
</p>
<button className="learn-more">LEARN MORE</button>
</div>
<div className="experience-cards" id="strategy">
<div className="experience-card-header">STRATEGY</div>
<p>
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Labore
commodi error sed, odit facere quibusdam exercitationem atque
assumenda nihil laudantium ipsum enim voluptatem. Reiciendis,
mollitia?
</p>
<button className="learn-more">LEARN MORE</button>
</div>
</div>
</div>
</div>
</>
);
};

export default Experience;
```

File: FirstFooter.jsx

```
import React from "react";

const FirstFooter = () => {
return (
<>
<div className="first-footer">
<div className="first-footer-container">
<p>
```

```
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Officiis
facere distinctio odio ipsum sint architecto.
</p>
<button className="learn">LEARN MORE</button>
</div>
</div>
</>
);
};

export default FirstFooter;
```

File: Header.jsx

```
import React from "react";

const Header = () => {
return (
<>
<div className="header">
<div className="minimal-design">
<h1 className="minimal-design-heading">We crush minimam design</h1>
<p className="a">
Lorem ipsum dolor sit amet consectetur adipisicing elit.
Reprehenderit delectus necessitatibus itaque fuga odio quidem
deserunt ullam cum magnam vero?
</p>
</div>
</div>
</>
);
};

export default Header;
```

File: Home.css

```
.nav {
display: flex;
flex-direction: row;
list-style: none;
align-items: center;
}
.nav * {
padding: 1rem;
}
.push {
margin-right: auto;
}
```

```
@media only screen and (max-width: 550px) {
.nav {
display: flex;
flex-direction: column;
list-style: none;
align-items: center;
}
.push {
margin: auto;
}
}

.header {
background: linear-gradient(rgba(0, 0, 0, 1), rgba(0, 0, 0, 0.9)),
url("house.jpg");
height: 60vh;
display: flex;
align-items: center;
padding: 7vw;
}

.minimal-design * {
color: white;
}

.minimal-design {
width: 30%;
}

.push {
}

@media only screen and (max-width: 500px) {
.experience {
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
}

.experience-body {
display: flex;
flex-direction: column !important;
width: 100vw;
}

.experience-header {
padding-left: 5vh;
}

.experience-child {
padding: 4vh !important;
}
```

```
#design {
text-align: left;
padding-left: 2vw !important;
}

#strategy {
text-align: left;
padding-right: 2vw !important;
}

.experience-cards{
padding-bottom: 5vh !important;
}

}

.experience {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
}

.experience-body {
display: flex;
flex-direction: row;
}

.experience-header {
padding-bottom: 5vh;
}

.experience-header h1 {
font-family: "Rowdies", cursive;
}

.experience-cards {
padding: 2vw;
}

#design {
padding-left: 0px;
}

#strategy {
padding-right: 0;
}

.experience-child {
padding: 20vh;
}
.experience-card-header {
font-size: 1.2rem;
font-weight: 700;
```

```
}

.learn-more {
background-color: black;
color: white;
padding: 0.8vw;
border-style: none;
}

.learn {
color:black;
padding: 0.5vw 1.5vw 0.5vw 1.5vw!important;
border-style: none;
}

.brands {
height: 50vh;
background: linear-gradient(rgba(0, 0, 0, 1), rgba(0, 0, 0, 0.9)),
url("house.jpg");
}

.stories {
padding-bottom: 10vh;
}
.first-footer {
background-color: aqua;
padding: 4vh;
}

@media only screen and (min-width: 500px) {
}

.second-footer {
background-color: rgb(216, 178, 223);
height: 15vh;
}
.brand-heading {
font-size: 2rem;
font-weight: 700;
}

.brand-headers {
color: white;
margin-bottom: 8vh;
}

.brands {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
}

.brands-container {
```

```
width: 80vw;
}

.brand-body {
display: flex;
flex-direction: row;
list-style: none;
justify-content: space-between;
}

.brand-body * {
color: white;
}

.stories {
display: flex;
justify-content: center;
}

.stories-container {
width: 75%;
padding: 10vh;
}

.stories-header {
margin-bottom: 10vh;
}

.stories-heading {
font-size: 700;
font-size: 2rem;
font-family: 'Rowdies', cursive;
}

.stories-body {
display: flex;
flex-direction: row;
justify-content: space-around;
}

@media only screen and (max-width: 768px) {
.stories-body {
display: flex;
flex-direction: column;
justify-content: center;
}
.stories-body * {
width: 100% !important;
}

.stories-container {
width: 100%;
padding: 2vh;
}
```

```
}

.body-first {
width: 45%;
}

.body-second {
width: 45%;
}

.body-image {
background: no-repeat 100% url("house.jpg");
height: 50vh;
background-size: 100% 100%;
}

.stories-body * .body-description-header {
padding-top: 3vh;
}

.stories-body * .body-description-date {
padding-top: 1vh;
}

.first-footer-container {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
}

.first-footer {
background: linear-gradient(
to right,
rgb(24, 89, 143),
rgba(92, 143, 219, 0.9)
);
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
}

.first-footer-container button {
border-style: none;
padding: 0.6vw;
}

.second-footer {
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
background: linear-gradient(rgba(0, 0, 0, 1), rgba(0, 0, 0, 0.9)),
```

```
url("house.jpg");
}

.second-footer-container {
width: 30vw;
display: flex;
flex-direction: column;
align-items: center;
}

.footer-first {
display: flex;
flex-direction: row;
list-style: none;
justify-content: space-around;
}

.second-footer * {
padding: 0.2vh;
color: white;
}

@media only screen and (max-width: 600px) {
.brand-body {
display: flex;
flex-direction: row;
justify-content: space-around;
flex-wrap: wrap;
}
}

.minimal-design-heading {
font-family: "Rowdies", cursive;
}

.brand-heading {
font-family: "Rowdies", cursive;
}
```

File: Home.jsx

```
import React from "react";
import "./Home.css";
import Navigation from "./Navigation/Navigation";
import Header from "./Header/Header";
import Experience from "./Experience/Experience";
import Brands from "./Brands/Brands";
import Stories from "./Stories/Stories";
import FirstFooter from "./FirstFooter/FirstFooter";
import SecondFooter from "./SecondFooter/SecondFooter";

const Home = () => {
```

```
return (
<>
<div className="container">
<Navigation />
<Header />
<Experience />
<Brands />
<Stories />
<FirstFooter />
<SecondFooter />
</div>
</>
);
};

export default Home;
```

File: Navigation.jsx

```
import React from 'react';
import { FiFacebook } from 'react-icons/fi';
import { AiOutlineSearch } from 'react-icons/ai';

const Navigation = () => {
return (
<>
<div className="nav">
<li className='push'><FiFacebook /></li>
<li>Sample</li>
<li>House</li>
<li>product</li>
<li>About</li>
<li>Contact</li>
<li>
<AiOutlineSearch/>
</li>
</div>
</>
)
}

export default Navigation
```

File: SecondFooter.jsx

```
import React from "react";
import { FiFacebook } from 'react-icons/fi';

const SecondFooter = () => {
return (
<>
```

```
<div className="second-footer">
<div className="second-footer-container">
<FiFacebook className="second-footer-image" />
<div className="footer-first">
<li>HelpDesk</li>
<li>change</li>
<li>powerered</li>
</div>
<div className="footer-first">
<li>lorem</li>
<li>change</li>
<li>contact</li>
</div>
</div>
</div>
</>
);
};

export default SecondFooter;
```

File: index.css

File: index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
<React.StrictMode>
<App />
</React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

Repository: calorie

File: README.md

```
## ![Dive logo](https://user-images.githubusercontent.com/424487/219708981-f0416526-ba48-4b01-b5b3-c0eb73362718.png) Dive
<!-- ![Company Logo](https://example.org) -->

| Octernship info  | Timelines and Stipend |
| ------------- | ------------- |
| Assignment Deadline  | 19th June 2023  |
| Octernship Duration  | 3rd July 2023 - 3rd October 2023 |
| Monthly Stipend  | $500 USD  |

## Assignment


# Write a REST API for the input of calories in Python

### Task Instructions
- API Users must be able to create an account and log in.
- All API calls must be authenticated.
- Implement at least three roles with different permission levels: a regular user would only be able to CRUD on their owned records, a user manager would be able to CRUD only users, and an admin would be able to CRUD all records and users.
- Each entry has a date, time, text, and number of calories.
- If the number of calories is not provided, the API should connect to a Calories API provider (for example, https://www.nutritionix.com) and try to get the number of calories for the entered meal.
- User setting ? Expected number of calories per day.
- Each entry should have an extra boolean field set to true if the total for that day is less than the expected number of calories per day, otherwise should be false.
- The API must be able to return data in the JSON format.
- The API should provide filter capabilities for all endpoints that return a list of elements, as well should be able to support pagination.
- Write unit and e2e tests.
- Use any *Python* web framework
- Use *SQLite* as the database

### Task Expectations
- API Design Best Practices
- Documentation of any assumptions or choices made and why
- Links as citation to any article / code referred to or used
- Unit tests covering the core calories logic
- Appropriate exception handling and error messages
- Code Quality - remove any unnecessary code, avoid large functions
- Good commit history - we won?t accept a repo with a single giant commit ????

### Task submission
Using the [GitHub Flow](https://docs.github.com/en/get-started/quickstart/github-flow#following-github-flo
```

w) for assignment submission
1. Creating a new branch
2. Raising a Pull Request for submission
3. Using GitHub Discussions to ask any relevant questions regarding the project
4. Final submission Checklist:
- [ ] SUBMISSION.md in the repository / PR, with:
- [ ] commands to set up the repo (dependencies etc.)
- [ ] commands to run the test suite
- [ ] commands to run the API server

Repository: cart-items

File: README.md

```
# Getting Started with Create React App

This project was bootstrapped with [Create React App](https://github.com/facebook/create-react-app).

## Available Scripts

In the project directory, you can run:

### `npm start`

Runs the app in the development mode.\
Open [http://localhost:3000](http://localhost:3000) to view it in your browser.

The page will reload when you make changes.\
You may also see any lint errors in the console.

### `npm test`

Launches the test runner in the interactive watch mode.\
See the section about [running tests](https://facebook.github.io/create-react-app/docs/running-tests) for more information.

### `npm run build`

Builds the app for production to the `build` folder.\
It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.\
Your app is ready to be deployed!

See the section about [deployment](https://facebook.github.io/create-react-app/docs/deployment) for more information.

### `npm run eject`

**Note: this is a one-way operation. Once you `eject`, you can't go back!**

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.
```

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

Learn More

You can learn more in the [Create React App documentation](https://facebook.github.io/create-react-app/docs/getting-started).

To learn React, check out the [React documentation](https://reactjs.org/).

Code Splitting

This section has moved here: https://facebook.github.io/create-react-app/docs/code-splitting

Analyzing the Bundle Size

This section has moved here: https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size

Making a Progressive Web App

This section has moved here: https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app

Advanced Configuration

This section has moved here: https://facebook.github.io/create-react-app/docs/advanced-configuration

Deployment

This section has moved here: https://facebook.github.io/create-react-app/docs/deployment

`npm run build` fails to minify

This section has moved here: https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify

File: index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta name="theme-color" content="#000000" />
<meta
name="description"
content="Web site created using create-react-app"
/>
<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
<!--
manifest.json provides metadata used when your web app is installed on a
user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
-->
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<!--
Notice the use of %PUBLIC_URL% in the tags above.
It will be replaced with the URL of the `public` folder during the build.
Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
work correctly both with client-side routing and a non-root public URL.
Learn how to configure a non-root public URL by running `npm run build`.
-->
<title>React App</title>
</head>
<body>
<noscript>You need to enable JavaScript to run this app.</noscript>
<div id="root"></div>
<!--
This HTML file is a template.
If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.
The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.
To create a production bundle, use `npm run build` or `yarn build`.
-->
</body>
</html>
```

File: robots.txt

```
# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
```

File: App.css

```
table{
width: 90vw !important;
margin:20px 10px 10px 10px;
box-shadow: 0px 5px 20px rgb(160, 65, 95);
border-radius: 20px;
}

.red-color{
background-color: red;
}
.green-color{
background-color:green;
}

td{
padding:1vw;
margin:1px;

}

.first{
background-color: red;
}

.second{
background-color: green;
}


* {
box-sizing: border-box;
}



.navbar {
overflow: hidden;
font-family: Arial, Helvetica, sans-serif;
}

.navbar a {
float: left;
font-size: 16px;
text-align: center;
/* padding: 14px 16px; */
text-decoration: none;
}

.dropdown {
float: left;
```

```
overflow: hidden;
}

.dropdown .dropbtn {

border: none;
outline: none;

padding: 5px 0px 0px 20px;
background-color: inherit;
font: inherit;
margin: 0;
}



.dropdown-content {
display: none;
position: absolute;
width: 100%;
left: 0;
z-index: 1;
}

.dropdown-content .header {
background: red;
padding: 16px;
color: white;
}

.dropdown:hover .dropdown-content {
display: block;
}

/* Create three equal columns that floats next to each other */
.column {
margin:0px 0px 0px 50px;
width: 10.33%;
padding: 1px;
background-color: #ccc;
height: 150px;
}

/* .column a {
float: none;
color: black;
padding: 16px;
text-decoration: none;
display: block;
text-align: left;
width:10px;
} */

.column a:hover {
```

```
background-color: #ddd;
}

/* Clear floats after the columns */
.row:after {
content: "";
display: table;
clear: both;
}

html{
overflow-x: hidden;
}

body{
overflow: hidden;
flex-direction: row;
display: flex;
justify-content: center;
align-items: center;
margin: 0px !important;
border-radius: 20px;
}

.header{
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
font-size: 3rem;
font-weight: 700;
color:coral;

}
```

File: App.js

```
import logo from './logo.svg';
import './App.css';

import TableContent from './components/Body';

function App() {


return (
<>
<div className="header">Tables</div>
<TableContent/>
</>
);
}
```

```
export default App;
```

File: App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
render(<App />);
const linkElement = screen.getByText(/learn react/i);
expect(linkElement).toBeInTheDocument();
});
```

File: Body.jsx

```
import React from 'react'
import { useEffect, useState } from 'react'
import Table from './Table';
import TableComponent from './TableComponent';

const TableContent = () => {
return (
<>
<body>
<div className="home-container">
<div className="serv">
<ul>
<TableComponent/>
</ul>
</div>
</div>
</body>
</>
)
}

export default TableContent
```

File: Table.jsx

```
import React from "react";

const Table = (props) => {
return (
<>
<tr>
<td>{props.item.first_name}</td>
<td>{props.item.last_name}</td>
<td>{props.item.email}</td>
```

```
<td>{props.item.gender}</td>
<td>{props.item.ip_address}</td>
<td>{props.item.airport_code}</td>
<td>{props.item.time}</td>

<td>
{props.item.status == true ? (
<div className="first">{props.item.status.toString()}</div>
) : (
<div className="second">{props.item.status.toString()}</div>
)}
</td>
<td>{props.item.mobile}</td>
<td>{props.item.area}</td>
<td>{props.item.show.toString()}</td>
<td> {props.item.status.toString()}</td>
</tr>
</>
);
};

export default Table;
```

File: TableComponent.jsx

```
import React from "react";
import Table from "./Table";
import { useEffect, useState } from "react";

const TableComponent = () => {
var datas = [
{
id: 1,
first_name: "Lonnie",
last_name: "O' Connell",
email: "loconnell0@hao123.com",
gender: "Male",
ip_address: "163.42.164.152",
airport_code: "OGO",
time: "7/10/2022",
status: true,
mobile: "707-462-9538",
area: "9 Coleman Trail",
show: false,
edit: false,
},
{
id: 2,
first_name: "Nikolos",
last_name: "Botler",
email: "nbotler1@wikipedia.org",
gender: "Male",
```

```
ip_address: "116.251.170.231",
airport_code: "CBB",
time: "12/25/2021",
status: false,
mobile: "240-316-9224",
area: "6545 Waxwing Road",
show: false,
edit: false,
},
{
id: 3,
first_name: "Burgess",
last_name: "Caddens",
email: "bcaddens2@accuweather.com",
gender: "Polygender",
ip_address: "126.177.211.243",
airport_code: "UKS",
time: "2/24/2022",
status: false,
mobile: "435-749-4095",
area: "49260 Golf Course Court",
show: false,
edit: true,
},
{
id: 4,
first_name: "Elissa",
last_name: "Lesslie",
email: "elesslie3@blog.com",
gender: "Female",
ip_address: "163.18.115.96",
airport_code: "SFE",
time: "3/13/2022",
status: false,
mobile: "877-992-0858",
area: "65014 Gulseth Trail",
show: false,
edit: false,
},
{
id: 5,
first_name: "Donaugh",
last_name: "Nusche",
email: "dnusche4@newsvine.com",
gender: "Male",
ip_address: "63.202.238.236",
airport_code: "VAW",
time: "6/2/2022",
status: true,
mobile: "419-777-9236",
area: "047 Anzinger Terrace",
show: true,
edit: false,
},
```

```
{
id: 6,
first_name: "Guillaume",
last_name: "Edgcumbe",
email: "gedgcumbe5@ycombinator.com",
gender: "Male",
ip_address: "134.249.161.241",
airport_code: "NOV",
time: "11/24/2022",
status: true,
mobile: "100-187-2648",
area: "262 Spohn Trail",
show: false,
edit: false,
},
{
id: 7,
first_name: "Berti",
last_name: "Coldbath",
email: "bcoldbath6@un.org",
gender: "Male",
ip_address: "130.30.122.132",
airport_code: "PHK",
time: "6/8/2022",
status: true,
mobile: "292-925-7639",
area: "5167 Surrey Junction",
show: true,
edit: true,
},
{
id: 8,
first_name: "Daune",
last_name: "Brumen",
email: "dbrumen7@si.edu",
gender: "Female",
ip_address: "250.92.132.133",
airport_code: "ZMH",
time: "8/17/2022",
status: false,
mobile: "467-673-4051",
area: "3076 Toban Avenue",
show: false,
edit: false,
},
{
id: 9,
first_name: "Aigneis",
last_name: "De Ruel",
email: "aderuel8@unicef.org",
gender: "Female",
ip_address: "99.140.129.80",
airport_code: "HUN",
time: "8/31/2022",
```

```
status: false,
mobile: "192-951-6084",
area: "6 Union Drive",
show: true,
edit: true,
},
{
id: 10,
first_name: "Khalil",
last_name: "McQuorkel",
email: "kmcquorkel9@zdnet.com",
gender: "Male",
ip_address: "172.193.34.60",
airport_code: "CSA",
time: "10/25/2022",
status: false,
mobile: "294-898-9568",
area: "708 Straubel Pass",
show: false,
edit: false,
},
{
id: 11,
first_name: "Xylina",
last_name: "Coom",
email: "xcooma@msu.edu",
gender: "Female",
ip_address: "65.171.59.54",
airport_code: "QRY",
time: "8/3/2022",
status: false,
mobile: "411-396-4207",
area: "789 Boyd Terrace",
show: true,
edit: false,
},
{
id: 12,
first_name: "Phillipp",
last_name: "Templeton",
email: "ptempletonb@dagondesign.com",
gender: "Male",
ip_address: "6.58.113.79",
airport_code: "YOA",
time: "5/9/2022",
status: true,
mobile: "978-733-1669",
area: "641 Eagan Crossing",
show: true,
edit: false,
},
{
id: 13,
first_name: "Janela",
```

```
last_name: "Harniman",
email: "jharnimanc@t-online.de",
gender: "Female",
ip_address: "180.22.90.230",
airport_code: "CVQ",
time: "9/24/2022",
status: true,
mobile: "179-908-3854",
area: "31078 Mallory Park",
show: true,
edit: true,
},
{
id: 14,
first_name: "Hamid",
last_name: "Dyster",
email: "hdysterd@cornell.edu",
gender: "Male",
ip_address: "61.88.24.2",
airport_code: "IKL",
time: "12/26/2021",
status: true,
mobile: "694-480-0182",
area: "8295 International Plaza",
show: true,
edit: false,
},
{
id: 15,
first_name: "Federica",
last_name: "Richel",
email: "frichele@ask.com",
gender: "Female",
ip_address: "183.113.111.234",
airport_code: "MGG",
time: "10/2/2022",
status: true,
mobile: "614-699-8295",
area: "267 Cordelia Court",
show: false,
edit: false,
},
{
id: 16,
first_name: "Rebe",
last_name: "Leipold",
email: "rleipoldf@bloglovin.com",
gender: "Polygender",
ip_address: "49.255.10.149",
airport_code: "RRK",
time: "7/10/2022",
status: true,
mobile: "445-992-5668",
area: "196 Truax Lane",
```

```
show: true,
edit: true,
},
{
id: 17,
first_name: "Rozina",
last_name: "Mattock",
email: "rmattockg@cisco.com",
gender: "Female",
ip_address: "190.112.193.84",
airport_code: "HTW",
time: "5/11/2022",
status: false,
mobile: "361-580-7808",
area: "7126 Bartelt Drive",
show: false,
edit: false,
},
{
id: 18,
first_name: "Raffarty",
last_name: "Kidwell",
email: "rkidwellh@washington.edu",
gender: "Male",
ip_address: "213.255.228.78",
airport_code: "FKN",
time: "5/10/2022",
status: false,
mobile: "435-771-2224",
area: "55527 Shopko Pass",
show: false,
edit: true,
},
{
id: 19,
first_name: "Ermina",
last_name: "Croucher",
email: "ecroucheri@prlog.org",
gender: "Polygender",
ip_address: "44.244.212.103",
airport_code: "CMH",
time: "11/24/2022",
status: true,
mobile: "929-930-9264",
area: "9 Algoma Parkway",
show: true,
edit: false,
},
{
id: 20,
first_name: "Virgie",
last_name: "Zuppa",
email: "vzuppaj@usa.gov",
gender: "Male",
```

```
ip_address: "103.60.243.234",
airport_code: "PUN",
time: "5/27/2022",
status: true,
mobile: "957-265-4315",
area: "1162 Bowman Center",
show: true,
edit: false,
},
{
id: 21,
first_name: "Eryn",
last_name: "L'Hommeau",
email: "elhommeauk@berkeley.edu",
gender: "Female",
ip_address: "115.252.199.72",
airport_code: "QVR",
time: "9/12/2022",
status: true,
mobile: "215-948-5791",
area: "74662 High Crossing Pass",
show: false,
edit: false,
},
{
id: 22,
first_name: "Marmaduke",
last_name: "Capaldo",
email: "mcapaldol@gmpg.org",
gender: "Male",
ip_address: "165.96.43.70",
airport_code: "VPY",
time: "4/10/2022",
status: false,
mobile: "433-519-9175",
area: "98913 Corscot Alley",
show: false,
edit: false,
},
{
id: 23,
first_name: "Rosalia",
last_name: "Humes",
email: "rhumesm@hhs.gov",
gender: "Female",
ip_address: "156.212.248.11",
airport_code: "SKR",
time: "2/1/2022",
status: true,
mobile: "799-773-0144",
area: "172 Buena Vista Parkway",
show: true,
edit: false,
},
```

```
{
id: 24,
first_name: "Vin",
last_name: "Arkle",
email: "varklen@time.com",
gender: "Male",
ip_address: "194.140.178.233",
airport_code: "GOT",
time: "5/6/2022",
status: false,
mobile: "994-130-4127",
area: "609 Dryden Trail",
show: true,
edit: true,
},
{
id: 25,
first_name: "Hollyanne",
last_name: "Schaumann",
email: "hschaumanno@nasa.gov",
gender: "Female",
ip_address: "32.105.42.36",
airport_code: "XAR",
time: "5/24/2022",
status: true,
mobile: "322-793-8279",
area: "5 Schmedeman Point",
show: false,
edit: false,
},
{
id: 26,
first_name: "Ransom",
last_name: "Tubritt",
email: "rtubrittp@purevolume.com",
gender: "Bigender",
ip_address: "41.51.38.166",
airport_code: "TEP",
time: "7/1/2022",
status: true,
mobile: "730-814-7105",
area: "48591 Bashford Park",
show: true,
edit: false,
},
{
id: 27,
first_name: "Ali",
last_name: "Pont",
email: "apontq@seattletimes.com",
gender: "Female",
ip_address: "232.12.5.48",
airport_code: "PVK",
time: "2/23/2022",
```

```
status: true,
mobile: "267-509-9238",
area: "165 Manley Trail",
show: true,
edit: false,
},
{
id: 28,
first_name: "Angus",
last_name: "Gresser",
email: "agresserr@squidoo.com",
gender: "Male",
ip_address: "92.222.167.226",
airport_code: "BGA",
time: "11/9/2022",
status: true,
mobile: "772-696-6954",
area: "711 Rusk Pass",
show: false,
edit: true,
},
{
id: 29,
first_name: "Liam",
last_name: "Waite",
email: "lwaites@earthlink.net",
gender: "Male",
ip_address: "237.211.29.167",
airport_code: "HRM",
time: "3/22/2022",
status: true,
mobile: "143-447-7237",
area: "54106 Eggendart Road",
show: true,
edit: true,
},
{
id: 30,
first_name: "Godwin",
last_name: "Pacheco",
email: "gpachecot@loc.gov",
gender: "Male",
ip_address: "15.220.247.191",
airport_code: "LDO",
time: "2/13/2022",
status: false,
mobile: "355-376-3513",
area: "94 Tennessee Junction",
show: false,
edit: false,
},
{
id: 31,
first_name: "Hussein",
```

```
last_name: "Bernot",
email: "hbernotu@theguardian.com",
gender: "Male",
ip_address: "34.100.163.232",
airport_code: "XIN",
time: "4/24/2022",
status: true,
mobile: "250-272-3899",
area: "09 Fremont Circle",
show: true,
edit: false,
},
{
id: 32,
first_name: "Findley",
last_name: "Alywin",
email: "falywinv@hexun.com",
gender: "Male",
ip_address: "162.106.223.235",
airport_code: "QGP",
time: "12/14/2021",
status: false,
mobile: "679-748-8145",
area: "5 Homewood Road",
show: false,
edit: false,
},
{
id: 33,
first_name: "Barth",
last_name: "Sacher",
email: "bsacherw@rambler.ru",
gender: "Male",
ip_address: "47.83.92.180",
airport_code: "HUV",
time: "6/1/2022",
status: false,
mobile: "992-102-6302",
area: "64 Johnson Crossing",
show: false,
edit: false,
},
{
id: 34,
first_name: "Ernestine",
last_name: "O'Neal",
email: "eonealx@patch.com",
gender: "Female",
ip_address: "241.66.234.15",
airport_code: "MTA",
time: "5/1/2022",
status: false,
mobile: "340-644-1469",
area: "6 Saint Paul Court",
```

```
show: true,
edit: false,
},
{
id: 35,
first_name: "Jandy",
last_name: "Titchen",
email: "jtitcheny@wikispaces.com",
gender: "Female",
ip_address: "221.234.160.167",
airport_code: "IEG",
time: "10/15/2022",
status: true,
mobile: "173-427-0443",
area: "55 Russell Circle",
show: true,
edit: false,
},
{
id: 36,
first_name: "Ellynn",
last_name: "Corzor",
email: "ecorzorz@usatoday.com",
gender: "Female",
ip_address: "82.17.103.11",
airport_code: "GYG",
time: "7/22/2022",
status: true,
mobile: "677-988-2055",
area: "71 Dunning Pass",
show: false,
edit: true,
},
{
id: 37,
first_name: "Orel",
last_name: "Beadle",
email: "obeadle10@digg.com",
gender: "Female",
ip_address: "225.76.215.255",
airport_code: "XRY",
time: "6/25/2022",
status: false,
mobile: "474-359-7710",
area: "9488 Valley Edge Hill",
show: false,
edit: true,
},
{
id: 38,
first_name: "North",
last_name: "Halms",
email: "nhalms11@squidoo.com",
gender: "Male",
```

```
ip_address: "53.38.184.72",
airport_code: "MFB",
time: "11/10/2022",
status: true,
mobile: "410-395-4287",
area: "466 Westerfield Point",
show: true,
edit: false,
},
{
id: 39,
first_name: "Walther",
last_name: "Richardson",
email: "wrichardson12@princeton.edu",
gender: "Male",
ip_address: "160.30.226.185",
airport_code: "TTO",
time: "8/13/2022",
status: true,
mobile: "814-511-1460",
area: "4 Texas Trail",
show: true,
edit: true,
},
{
id: 40,
first_name: "Blancha",
last_name: "Lefridge",
email: "blefridge13@biblegateway.com",
gender: "Female",
ip_address: "164.20.75.197",
airport_code: "DKI",
time: "10/29/2022",
status: true,
mobile: "988-983-2725",
area: "844 Washington Road",
show: true,
edit: true,
},
{
id: 41,
first_name: "Matti",
last_name: "Boncore",
email: "mboncore14@addtoany.com",
gender: "Female",
ip_address: "79.211.151.234",
airport_code: "SHS",
time: "1/13/2022",
status: true,
mobile: "231-109-5204",
area: "45 Hansons Hill",
show: true,
edit: false,
},
```

```
{
id: 42,
first_name: "Erik",
last_name: "Richt",
email: "ericht15@qq.com",
gender: "Male",
ip_address: "35.39.154.14",
airport_code: "NDG",
time: "1/4/2022",
status: true,
mobile: "101-730-8823",
area: "1161 Forster Park",
show: true,
edit: true,
},
{
id: 43,
first_name: "Sidonnie",
last_name: "Curson",
email: "scurson16@google.co.jp",
gender: "Female",
ip_address: "94.48.147.216",
airport_code: "OMS",
time: "6/17/2022",
status: true,
mobile: "925-291-3451",
area: "42024 Scofield Street",
show: true,
edit: false,
},
{
id: 44,
first_name: "Ermentrude",
last_name: "Ordelt",
email: "eordelt17@hud.gov",
gender: "Female",
ip_address: "173.97.103.242",
airport_code: "WUU",
time: "1/23/2022",
status: true,
mobile: "636-258-4576",
area: "7625 Sundown Place",
show: true,
edit: true,
},
{
id: 45,
first_name: "Amos",
last_name: "Redd",
email: "aredd18@mail.ru",
gender: "Male",
ip_address: "100.161.79.55",
airport_code: "TIP",
time: "9/27/2022",
```

```
status: true,
mobile: "813-314-0163",
area: "0 Sugar Avenue",
show: true,
edit: true,
},
{
id: 46,
first_name: "Shaughn",
last_name: "Russon",
email: "srusson19@squidoo.com",
gender: "Male",
ip_address: "142.240.189.61",
airport_code: "MAD",
time: "1/18/2022",
status: false,
mobile: "176-722-4156",
area: "0392 Derek Alley",
show: true,
edit: false,
},
{
id: 47,
first_name: "Frans",
last_name: "Albiston",
email: "falbiston1a@phpbb.com",
gender: "Male",
ip_address: "131.241.104.200",
airport_code: "YQR",
time: "12/18/2021",
status: false,
mobile: "833-657-5677",
area: "18845 Thackeray Street",
show: true,
edit: false,
},
{
id: 48,
first_name: "Priscilla",
last_name: "Buckley",
email: "pbuckley1b@cornell.edu",
gender: "Female",
ip_address: "249.116.245.108",
airport_code: "AXS",
time: "1/29/2022",
status: false,
mobile: "904-986-5580",
area: "43 Merry Crossing",
show: true,
edit: true,
},
{
id: 49,
first_name: "Anastasie",
```

```
last_name: "Ekins",
email: "aekins1c@php.net",
gender: "Female",
ip_address: "249.167.17.233",
airport_code: "IAS",
time: "11/17/2022",
status: true,
mobile: "258-170-8566",
area: "65 Nelson Place",
show: false,
edit: false,
},
{
id: 50,
first_name: "Dunn",
last_name: "Brocking",
email: "dbrocking1d@amazon.co.jp",
gender: "Male",
ip_address: "100.114.28.23",
airport_code: "CBG",
time: "8/8/2022",
status: true,
mobile: "182-532-5686",
area: "350 Debs Circle",
show: false,
edit: true,
},
{
id: 51,
first_name: "Lorinda",
last_name: "Streets",
email: "lstreets1e@bbc.co.uk",
gender: "Female",
ip_address: "247.186.90.23",
airport_code: "EOZ",
time: "5/11/2022",
status: false,
mobile: "245-746-3422",
area: "5 Oak Valley Parkway",
show: false,
edit: true,
},
{
id: 52,
first_name: "Michaeline",
last_name: "Settle",
email: "msettle1f@bloglovin.com",
gender: "Female",
ip_address: "215.220.202.208",
airport_code: "TET",
time: "11/9/2022",
status: true,
mobile: "161-540-1638",
area: "08 Green Ridge Hill",
```

```
show: false,
edit: false,
},
{
id: 53,
first_name: "Laney",
last_name: "Ledur",
email: "lledur1g@unesco.org",
gender: "Male",
ip_address: "70.125.49.60",
airport_code: "KDN",
time: "10/17/2022",
status: true,
mobile: "324-630-3034",
area: "6 Almo Plaza",
show: false,
edit: true,
},
{
id: 54,
first_name: "Fianna",
last_name: "Curcher",
email: "fcurcher1h@npr.org",
gender: "Female",
ip_address: "116.75.35.16",
airport_code: "HPT",
time: "10/1/2022",
status: false,
mobile: "761-744-9586",
area: "758 Acker Pass",
show: true,
edit: true,
},
{
id: 55,
first_name: "Dmitri",
last_name: "Thunderman",
email: "dthunderman1i@economist.com",
gender: "Male",
ip_address: "7.8.196.177",
airport_code: "BVM",
time: "5/23/2022",
status: false,
mobile: "842-238-7525",
area: "6 Old Gate Parkway",
show: false,
edit: true,
},
{
id: 56,
first_name: "Karlene",
last_name: "Copin",
email: "kcopin1j@free.fr",
gender: "Female",
```

```
ip_address: "171.210.144.216",
airport_code: "PCB",
time: "1/10/2022",
status: false,
mobile: "187-665-1483",
area: "82909 Orin Junction",
show: true,
edit: false,
},
{
id: 57,
first_name: "Somerset",
last_name: "Wheble",
email: "swheble1k@hao123.com",
gender: "Male",
ip_address: "177.144.246.201",
airport_code: "HIB",
time: "6/25/2022",
status: false,
mobile: "711-245-3109",
area: "93 Anhalt Center",
show: false,
edit: false,
},
{
id: 58,
first_name: "Julita",
last_name: "Kenward",
email: "jkenward1l@51.la",
gender: "Female",
ip_address: "62.221.145.159",
airport_code: "ORV",
time: "3/14/2022",
status: false,
mobile: "151-237-4141",
area: "32 Arizona Junction",
show: true,
edit: false,
},
{
id: 59,
first_name: "Izaak",
last_name: "Grennan",
email: "igrennan1m@studiopress.com",
gender: "Male",
ip_address: "216.6.178.199",
airport_code: "KBN",
time: "3/24/2022",
status: false,
mobile: "500-964-5625",
area: "10 Brown Plaza",
show: false,
edit: true,
},
```

```
{
id: 60,
first_name: "Jozef",
last_name: "Collings",
email: "jcollings1n@unblog.fr",
gender: "Male",
ip_address: "106.178.77.44",
airport_code: "PYR",
time: "12/15/2021",
status: false,
mobile: "763-680-7533",
area: "5924 Haas Hill",
show: false,
edit: true,
},
{
id: 61,
first_name: "Si",
last_name: "Hehir",
email: "shehir1o@friendfeed.com",
gender: "Male",
ip_address: "21.215.94.119",
airport_code: "HBG",
time: "7/3/2022",
status: false,
mobile: "980-579-1417",
area: "26346 Merrick Place",
show: true,
edit: true,
},
{
id: 62,
first_name: "Nilson",
last_name: "Alldread",
email: "nalldread1p@creativecommons.org",
gender: "Male",
ip_address: "86.207.250.119",
airport_code: "SRE",
time: "8/4/2022",
status: false,
mobile: "832-787-2127",
area: "37 Kedzie Parkway",
show: false,
edit: false,
},
{
id: 63,
first_name: "Dorothee",
last_name: "Gelardi",
email: "dgelardi1q@sphinn.com",
gender: "Female",
ip_address: "114.152.230.46",
airport_code: "UBP",
time: "11/28/2022",
```

```
status: false,
mobile: "372-284-8952",
area: "144 Glacier Hill Hill",
show: false,
edit: true,
},
{
id: 64,
first_name: "Creighton",
last_name: "Langlois",
email: "clanglois1r@php.net",
gender: "Male",
ip_address: "47.165.116.4",
airport_code: "LBM",
time: "4/5/2022",
status: true,
mobile: "390-580-9920",
area: "4 Kropf Road",
show: true,
edit: false,
},
{
id: 65,
first_name: "Mac",
last_name: "de Banke",
email: "mdebanke1s@yellowbook.com",
gender: "Male",
ip_address: "243.217.42.151",
airport_code: "ONG",
time: "10/16/2022",
status: false,
mobile: "500-513-8879",
area: "8 Spohn Junction",
show: true,
edit: false,
},
{
id: 66,
first_name: "Othilie",
last_name: "Chaucer",
email: "ochaucer1t@mozilla.org",
gender: "Genderqueer",
ip_address: "185.207.93.176",
airport_code: "IRC",
time: "12/29/2021",
status: false,
mobile: "757-566-5544",
area: "88 Amoth Hill",
show: true,
edit: false,
},
{
id: 67,
first_name: "Salvidor",
```

```
last_name: "Caze",
email: "scaze1u@nationalgeographic.com",
gender: "Male",
ip_address: "209.218.233.62",
airport_code: "MCE",
time: "7/4/2022",
status: false,
mobile: "913-866-8369",
area: "806 7th Road",
show: false,
edit: false,
},
{
id: 68,
first_name: "Corbet",
last_name: "Luard",
email: "cluard1v@stumbleupon.com",
gender: "Male",
ip_address: "75.177.244.221",
airport_code: "CBI",
time: "8/22/2022",
status: false,
mobile: "382-644-6490",
area: "4 Merrick Trail",
show: false,
edit: false,
},
{
id: 69,
first_name: "Roddie",
last_name: "Golds",
email: "rgolds1w@topsy.com",
gender: "Male",
ip_address: "145.216.11.99",
airport_code: "KGO",
time: "5/27/2022",
status: true,
mobile: "123-905-3792",
area: "89727 Marcy Hill",
show: true,
edit: false,
},
{
id: 70,
first_name: "Wells",
last_name: "St. Pierre",
email: "wstpierre1x@mozilla.com",
gender: "Male",
ip_address: "215.185.68.161",
airport_code: "SCC",
time: "4/5/2022",
status: true,
mobile: "937-474-2108",
area: "5 Graedel Point",
```

```
show: false,
edit: true,
},
{
id: 71,
first_name: "Branden",
last_name: "Alvarado",
email: "balvarado1y@cdbaby.com",
gender: "Male",
ip_address: "63.6.246.90",
airport_code: "SPW",
time: "4/29/2022",
status: true,
mobile: "819-554-7746",
area: "2140 Cody Crossing",
show: true,
edit: false,
},
{
id: 72,
first_name: "Garvin",
last_name: "Berecloth",
email: "gberecloth1z@google.ru",
gender: "Male",
ip_address: "92.239.205.22",
airport_code: "IXG",
time: "8/30/2022",
status: true,
mobile: "616-114-9901",
area: "7372 Killdeer Lane",
show: true,
edit: false,
},
{
id: 73,
first_name: "Tomi",
last_name: "Kilgrew",
email: "tkilgrew20@unesco.org",
gender: "Female",
ip_address: "193.114.14.47",
airport_code: "CHG",
time: "5/17/2022",
status: true,
mobile: "830-293-0843",
area: "682 Killdeer Trail",
show: false,
edit: true,
},
{
id: 74,
first_name: "Bettye",
last_name: "Carnson",
email: "bcarnson21@google.cn",
gender: "Female",
```

```
ip_address: "106.14.138.49",
airport_code: "UPL",
time: "11/6/2021",
status: false,
mobile: "783-207-0588",
area: "41 Mariners Cove Hill",
show: true,
edit: true,
},
{
id: 75,
first_name: "Sholom",
last_name: "Dumper",
email: "sdumper22@scribd.com",
gender: "Male",
ip_address: "62.47.33.191",
airport_code: "WPK",
time: "4/27/2022",
status: false,
mobile: "593-833-6062",
area: "12822 Springs Circle",
show: false,
edit: true,
},
{
id: 76,
first_name: "Giorgia",
last_name: "Newrick",
email: "gnewrick23@rakuten.co.jp",
gender: "Female",
ip_address: "177.164.129.187",
airport_code: "NIO",
time: "7/18/2022",
status: true,
mobile: "981-341-7954",
area: "80 Brentwood Street",
show: true,
edit: false,
},
{
id: 77,
first_name: "Reece",
last_name: "Killingsworth",
email: "rkillingsworth24@pcworld.com",
gender: "Male",
ip_address: "33.254.193.3",
airport_code: "SFX",
time: "4/8/2022",
status: false,
mobile: "243-295-6548",
area: "02880 Portage Pass",
show: false,
edit: true,
},
```

```
{
id: 78,
first_name: "Carly",
last_name: "Boich",
email: "cboich25@netvibes.com",
gender: "Female",
ip_address: "184.213.163.222",
airport_code: "CNH",
time: "4/1/2022",
status: false,
mobile: "856-426-0826",
area: "45 Oak Valley Hill",
show: true,
edit: true,
},
{
id: 79,
first_name: "Cos",
last_name: "Corten",
email: "ccorten26@imdb.com",
gender: "Male",
ip_address: "124.1.61.158",
airport_code: "LPT",
time: "2/16/2022",
status: false,
mobile: "613-368-3194",
area: "07 Debra Drive",
show: false,
edit: false,
},
{
id: 80,
first_name: "Franny",
last_name: "Tennewell",
email: "ftennewell27@a8.net",
gender: "Bigender",
ip_address: "57.38.212.112",
airport_code: "BQL",
time: "4/3/2022",
status: true,
mobile: "908-972-2555",
area: "8586 Mitchell Crossing",
show: true,
edit: false,
},
{
id: 81,
first_name: "North",
last_name: "Jays",
email: "njays28@furl.net",
gender: "Male",
ip_address: "167.48.13.116",
airport_code: "TGA",
time: "4/10/2022",
```

```
status: true,
mobile: "844-146-8619",
area: "8473 Sullivan Point",
show: false,
edit: true,
},
{
id: 82,
first_name: "Chelsae",
last_name: "Caw",
email: "ccaw29@spotify.com",
gender: "Genderqueer",
ip_address: "94.90.71.179",
airport_code: "BXB",
time: "12/14/2021",
status: false,
mobile: "356-484-7734",
area: "97 John Wall Place",
show: true,
edit: true,
},
{
id: 83,
first_name: "Granny",
last_name: "Looker",
email: "glooker2a@technorati.com",
gender: "Male",
ip_address: "113.86.208.226",
airport_code: "KNR",
time: "11/14/2021",
status: true,
mobile: "470-804-2380",
area: "95 Bayside Junction",
show: true,
edit: true,
},
{
id: 84,
first_name: "Velvet",
last_name: "Blethin",
email: "vblethin2b@nih.gov",
gender: "Female",
ip_address: "51.40.20.112",
airport_code: "BRO",
time: "8/4/2022",
status: true,
mobile: "322-767-1391",
area: "4 Charing Cross Way",
show: true,
edit: true,
},
{
id: 85,
first_name: "Hillyer",
```

```
last_name: "Trippitt",
email: "htrippitt2c@constantcontact.com",
gender: "Male",
ip_address: "26.182.137.201",
airport_code: "LEV",
time: "1/15/2022",
status: false,
mobile: "779-350-6159",
area: "1951 Hintze Street",
show: true,
edit: true,
},
{
id: 86,
first_name: "Annora",
last_name: "Teeney",
email: "ateeney2d@nationalgeographic.com",
gender: "Female",
ip_address: "202.175.87.66",
airport_code: "CGP",
time: "11/27/2021",
status: true,
mobile: "156-704-2070",
area: "73869 Mendota Avenue",
show: true,
edit: false,
},
{
id: 87,
first_name: "Amery",
last_name: "Vasyutin",
email: "avasyutin2e@google.com.au",
gender: "Male",
ip_address: "232.50.173.16",
airport_code: "HTS",
time: "11/26/2021",
status: true,
mobile: "636-716-7218",
area: "5342 New Castle Drive",
show: true,
edit: false,
},
{
id: 88,
first_name: "Dougie",
last_name: "Iiannone",
email: "diiannone2f@vinaora.com",
gender: "Male",
ip_address: "78.32.181.142",
airport_code: "CKE",
time: "8/22/2022",
status: false,
mobile: "525-204-3241",
area: "11857 Monument Pass",
```

```
show: true,
edit: false,
},
{
id: 89,
first_name: "Wainwright",
last_name: "Coombs",
email: "wcoombs2g@netscape.com",
gender: "Male",
ip_address: "233.30.35.81",
airport_code: "ZUM",
time: "4/29/2022",
status: true,
mobile: "403-662-3636",
area: "0 Northridge Avenue",
show: false,
edit: false,
},
{
id: 90,
first_name: "Alexandrina",
last_name: "Farlow",
email: "afarlow2h@delicious.com",
gender: "Female",
ip_address: "139.119.14.201",
airport_code: "KDN",
time: "12/19/2021",
status: true,
mobile: "646-205-1116",
area: "74 Shelley Center",
show: false,
edit: false,
},
{
id: 91,
first_name: "Costa",
last_name: "Ruske",
email: "cruske2i@cdbaby.com",
gender: "Male",
ip_address: "215.25.144.171",
airport_code: "CUU",
time: "10/11/2022",
status: true,
mobile: "314-297-8121",
area: "58 Sutteridge Plaza",
show: false,
edit: true,
},
{
id: 92,
first_name: "Karil",
last_name: "Blamey",
email: "kblamey2j@yale.edu",
gender: "Female",
```

```
ip_address: "108.132.163.130",
airport_code: "XNG",
time: "9/12/2022",
status: false,
mobile: "226-892-8227",
area: "3793 Continental Alley",
show: false,
edit: false,
},
{
id: 93,
first_name: "Aldwin",
last_name: "Plenty",
email: "aplenty2k@seesaa.net",
gender: "Male",
ip_address: "43.246.252.192",
airport_code: "ADC",
time: "11/5/2022",
status: false,
mobile: "578-155-5426",
area: "1082 Charing Cross Hill",
show: true,
edit: true,
},
{
id: 94,
first_name: "Gwendolen",
last_name: "Duerdin",
email: "gduerdin2l@umich.edu",
gender: "Female",
ip_address: "222.199.152.229",
airport_code: "TTE",
time: "12/2/2021",
status: false,
mobile: "497-776-9498",
area: "42 Mccormick Place",
show: true,
edit: true,
},
{
id: 95,
first_name: "Camellia",
last_name: "Rhymer",
email: "crhymer2m@economist.com",
gender: "Female",
ip_address: "124.191.75.110",
airport_code: "LOW",
time: "7/6/2022",
status: true,
mobile: "817-929-8305",
area: "6 Badeau Pass",
show: false,
edit: true,
},
```

```
{
id: 96,
first_name: "Webb",
last_name: "Spruce",
email: "wspruce2n@washingtonpost.com",
gender: "Male",
ip_address: "129.200.134.134",
airport_code: "HNH",
time: "10/5/2022",
status: false,
mobile: "743-482-1273",
area: "21143 Schmedeman Way",
show: false,
edit: false,
},
{
id: 97,
first_name: "Marv",
last_name: "Cragg",
email: "mcragg2o@mlb.com",
gender: "Male",
ip_address: "100.238.64.46",
airport_code: "AXV",
time: "10/8/2022",
status: true,
mobile: "480-664-7539",
area: "0308 Dawn Terrace",
show: false,
edit: true,
},
{
id: 98,
first_name: "Saunderson",
last_name: "Weston",
email: "sweston2p@pbs.org",
gender: "Male",
ip_address: "236.130.198.148",
airport_code: "MIZ",
time: "6/28/2022",
status: false,
mobile: "823-450-9719",
area: "1 Ohio Place",
show: true,
edit: false,
},
{
id: 99,
first_name: "Lionel",
last_name: "Beedie",
email: "lbeedie2q@chron.com",
gender: "Male",
ip_address: "43.183.53.122",
airport_code: "TBH",
time: "11/7/2022",
```

```
status: false,
mobile: "717-482-7772",
area: "55 Sugar Plaza",
show: false,
edit: true,
},
{
id: 100,
first_name: "Reuben",
last_name: "Tesyro",
email: "rtesyro2r@fotki.com",
gender: "Polygender",
ip_address: "124.233.62.26",
airport_code: "CMU",
time: "1/8/2022",
status: true,
mobile: "287-143-2240",
area: "3996 Vera Street",
show: false,
edit: false,
},
];

var [data, setData] = useState([]);

useEffect(() => {
setData(datas);
}, []);
const accend = (e) => {
datas = datas.sort((a, b) => {
if (a.first_name < b.first_name) {
return -1;
}
});
console.log(e.target.id);
setData(datas);
};
const decend = (e) => {
datas = datas.sort((a, b) => {
if (a.first_name > b.first_name) {
return -1;
}
});
console.log(e.target.id);
setData(datas);
};

return (
<>
<table>
<th>
<div className="navbar">
<div className="dropdown">
<button className="dropbtn">
```

```
First Name
<i className="fa fa-caret-down"></i>
</button>
<div className="dropdown-content">
<div className="column">
<button className="btn" id1="a.first_name" onClick={accend}>
ascend order
</button>
<button className="btn" id="b.first_name" onClick={decend}>
Decend order
</button>
</div>
</div>
</div>
</div>
</th>
<th>Last Name</th>
<th>Email</th>
<th>Gender</th>
<th>IP address</th>
<th>airport code</th>
<th>Time</th>
<th>Status</th>
<th>Mobile</th>
<th>Area</th>
<th>Show</th>
<th>Edit</th>
{data.map((items, idx) => {
return (
<>
<Table item={items} />
</>
);
})}
</table>
</>
);
};

export default TableComponent;
```

File: index.css

File: index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
```

```
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
<React.StrictMode>
<App />
</React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

File: reportWebVitals.js

```
const reportWebVitals = onPerfEntry => {
if (onPerfEntry && onPerfEntry instanceof Function) {
import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
getCLS(onPerfEntry);
getFID(onPerfEntry);
getFCP(onPerfEntry);
getLCP(onPerfEntry);
getTTFB(onPerfEntry);
});
}
};

export default reportWebVitals;
```

Repository: ci-cd-learning

File: README.md

```
# ci-cd-learning
```

File: __init__.py

File: asgi.py

```
"""
ASGI config for backend project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'backend.settings')

application = get_asgi_application()
```

File: settings.py

```
"""
Django settings for backend project.

Generated by 'django-admin startproject' using Django 4.2.1.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.2/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-q1a6$@f2dpn*w-4a24+vv9qm=126kkhrj*#ine%x+*=__7)&$3'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'corsheaders',
'rest_framework',
'homepage',
]

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
'corsheaders.middleware.CorsMiddleware',
]

ROOT_URLCONF = 'backend.urls'

TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
```

```
]

WSGI_APPLICATION = 'backend.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.2/ref/settings/#databases

DATABASES = {
'default': {
'ENGINE': 'django.db.backends.sqlite3',
'NAME': BASE_DIR / 'db.sqlite3',
}
}


# Password validation
# https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
{
'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]


# Internationalization
# https://docs.djangoproject.com/en/4.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field
```

```
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'


CORS_ORIGIN_WHITELIST = [
'http://localhost:3000'
]
```

File: urls.py

```
"""
URL configuration for backend project.

The `urlpatterns` list routes URLs to views. For more information please see:
https://docs.djangoproject.com/en/4.2/topics/http/urls/
Examples:
Function views
1. Add an import:  from my_app import views
2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
1. Add an import:  from other_app.views import Home
2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include
from rest_framework import routers
from homepage import views
# from homepage.views import message_view, homepage

router=routers.DefaultRouter()
# router.register(r"homepages", home_page_view, 'homepage')


urlpatterns = [
path('admin/', admin.site.urls),
path('', include('homepage.urls')),
]
```

File: wsgi.py

```
"""
WSGI config for backend project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/howto/deployment/wsgi/
"""
```

```
import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'backend.settings')

application = get_wsgi_application()
```

File: admin.py

```
from django.contrib import admin

from .models import Messages, EmailIds

# Register your models here.

class MessagesAdmin(admin.ModelAdmin):
list_display=("first_name", "last_name", "email_id","message_box")

class EmailIdsAdmin(admin.ModelAdmin):
list_display=("email_id",)

admin.site.register(Messages, MessagesAdmin)
admin.site.register(EmailIds, EmailIdsAdmin)
```

File: apps.py

```
from django.apps import AppConfig


class HomepageConfig(AppConfig):
default_auto_field = 'django.db.models.BigAutoField'
name = 'homepage'
```

File: 0001_initial.py

```
# Generated by Django 4.2.1 on 2023-05-31 05:18

from django.db import migrations, models


class Migration(migrations.Migration):

initial = True

dependencies = [
]
```

```
operations = [
migrations.CreateModel(
name='HomePage',
fields=[
('id',    models.BigAutoField(auto_created=True,    primary_key=True,    serialize=False,
verbose_name='ID')),
('name', models.CharField(max_length=120)),
('title', models.CharField(max_length=50)),
],
),
]
```

File: 0002_homepage_email_address.py

```
# Generated by Django 4.2.1 on 2023-05-31 05:55

from django.db import migrations, models


class Migration(migrations.Migration):

dependencies = [
('homepage', '0001_initial'),
]

operations = [
migrations.AddField(
model_name='homepage',
name='email_address',
field=models.EmailField(default='none', max_length=120),
preserve_default=False,
),
]
```

File: 0003_remove_homepage_title_homepage_concern.py

```
# Generated by Django 4.2.1 on 2023-05-31 06:01

from django.db import migrations, models


class Migration(migrations.Migration):

dependencies = [
('homepage', '0002_homepage_email_address'),
]

operations = [
migrations.RemoveField(
```

```
model_name='homepage',
name='title',
),
migrations.AddField(
model_name='homepage',
name='concern',
field=models.CharField(default='None', max_length=256),
preserve_default=False,
),
]
```

File: 0004_rename_email_address_homepage_email_id_and_more.py

---

```
# Generated by Django 4.2.1 on 2023-05-31 09:20

from django.db import migrations, models


class Migration(migrations.Migration):

dependencies = [
('homepage', '0003_remove_homepage_title_homepage_concern'),
]

operations = [
migrations.RenameField(
model_name='homepage',
old_name='email_address',
new_name='email_id',
),
migrations.RenameField(
model_name='homepage',
old_name='name',
new_name='first_name',
),
migrations.RenameField(
model_name='homepage',
old_name='concern',
new_name='message_box',
),
migrations.AddField(
model_name='homepage',
name='last_name',
field=models.CharField(default='None', max_length=120),
preserve_default=False,
),
]
```

File: 0005_emailids.py

```
# Generated by Django 4.2.1 on 2023-06-03 03:46

from django.db import migrations, models


class Migration(migrations.Migration):

dependencies = [
('homepage', '0004_rename_email_address_homepage_email_id_and_more'),
]

operations = [
migrations.CreateModel(
name='EmailIds',
fields=[
('id',    models.BigAutoField(auto_created=True,    primary_key=True,    serialize=False,
verbose_name='ID')),
('email_id', models.CharField(max_length=120)),
],
),
]
```

File: 0006_rename_homepage_messages.py

```
# Generated by Django 4.2.1 on 2023-06-03 04:43

from django.db import migrations


class Migration(migrations.Migration):

dependencies = [
('homepage', '0005_emailids'),
]

operations = [
migrations.RenameModel(
old_name='HomePage',
new_name='Messages',
),
]
```

File: models.py

```
from django.db import models

# Create your models here.


class Messages(models.Model):
```

```
first_name = models.CharField(max_length=120)
last_name = models.CharField(max_length=120)
email_id=models.EmailField(max_length=120)
message_box=models.CharField(max_length=256)

def __str__(self):
return "Data of "+ self.first_name + "Email address: " +self.email_id

class EmailIds(models.Model):
email_id=models.CharField(max_length=120)

def __str__(self):
return self.email_id
```

File: serializers.py

```
from rest_framework import serializers
from .models import Messages, EmailIds

class MessageSerializer(serializers.ModelSerializer):
class Meta:
model=Messages
fields=("id","first_name","last_name","email_id","message_box")

class EmailSerializer(serializers.ModelSerializer):
class Meta:
model=EmailIds
fields=("email_id",)
```

File: tests.py

```
from django.test import TestCase

# Create your tests here.
```

File: manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
"""Run administrative tasks."""
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'backend.settings')
try:
```

```
from django.core.management import execute_from_command_line
except ImportError as exc:
raise ImportError(
"Couldn't import Django. Are you sure it's installed and "
"available on your PYTHONPATH environment variable? Did you "
"forget to activate a virtual environment?"
) from exc
execute_from_command_line(sys.argv)


if __name__ == '__main__':
main()
```