

```
=====
File: package.json
=====
```

```
{
  "name": "jobify",
  "version": "1.0.0",
  "description": "[Jobify](https://mern-jobify-v2.onrender.com/)",
  "main": "index.js",
  "type": "module",
  "scripts": {
    "setup-project": "npm i && cd client && npm i",
    "setup-production-app": "npm i && cd client && npm i && npm run build ",
    "server": "nodemon server",
    "client": "cd client && npm run dev",
    "dev": "concurrently --kill-others-on-fail \"npm run server\" \"npm run client\""
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "cloudinary": "^1.37.3",
    "concurrently": "^8.0.1",
    "cookie-parser": "^1.4.6",
    "datauri": "^4.1.0",
    "dayjs": "^1.11.7",
```

```
"dotenv": "^16.0.3",
"express": "^4.18.2",
"express-async-errors": "^3.1.1",
"express-mongo-sanitize": "^2.2.0",
"express-rate-limit": "^6.8.0",
"express-validator": "^7.0.1",
"helmet": "^7.0.0",
"http-status-codes": "^2.2.0",
"jsonwebtoken": "^9.0.0",
"mongoose": "^7.0.5",
"morgan": "^1.10.0",
"multer": "^1.4.5-lts.1",
"nanoid": "^4.0.2",
"nodemon": "^2.0.22"
}
}
```

=====

File: populate.js

=====

```
import { readFile } from 'fs/promises';
import mongoose from 'mongoose';
import dotenv from 'dotenv';
dotenv.config();
```

```

import Job from './models/JobModel.js';
import User from './models/UserModel.js';

try {
  await mongoose.connect(process.env.MONGO_URL);
  const user = await User.findOne({ email: 'john@gmail.com' });
  const jsonJobs = JSON.parse(
    await readFile(new URL('./utils/mockData.json', import.meta.url))
  );
  const jobs = jsonJobs.map((job) => {
    return { ...job, createdBy: user._id };
  });
  await Job.deleteMany({ createdBy: user._id });
  await Job.create(jobs);
  console.log('Success!!!');
  process.exit(0);
} catch (error) {
  console.log(error);
  process.exit(1);
}

```

```
=====
```

File: server.js

```
=====
```

```
import 'express-async-errors';
```

```
import * as dotenv from 'dotenv';
dotenv.config();
import express from 'express';
const app = express();
import morgan from 'morgan';
import mongoose from 'mongoose';
import cookieParser from 'cookie-parser';
import cloudinary from 'cloudinary';
import helmet from 'helmet';
import mongoSanitize from 'express-mongo-sanitize';

// routers
import jobRouter from './routes/jobRouter.js';
import authRouter from './routes/authRouter.js';
import userRouter from './routes/userRouter.js';

// public
import { dirname } from 'path';
import { fileURLToPath } from 'url';
import path from 'path';

// middleware
import errorHandlerMiddleware from './middleware/errorHandlerMiddleware.js';
import { authenticateUser } from './middleware/authMiddleware.js';

cloudinary.config({
  cloud_name: process.env.CLOUD_NAME,
```

```
    api_key: process.env.CLOUD_API_KEY,  
    api_secret: process.env.CLOUD_API_SECRET,  
  });
```

```
const __dirname = dirname(fileURLToPath(import.meta.url));  
if (process.env.NODE_ENV === 'development') {  
  app.use(morgan('dev'));  
}  
app.use(express.static(path.resolve(__dirname, './client/dist')));  
app.use(cookieParser());  
app.use(express.json());  
app.use(helmet());  
app.use(mongoSanitize());
```

```
app.get('/', (req, res) => {  
  res.send('Hello World');  
});
```

```
app.get('/api/v1/test', (req, res) => {  
  res.json({ msg: 'test route' });  
});
```

```
app.use('/api/v1/jobs', authenticateUser, jobRouter);  
app.use('/api/v1/users', authenticateUser, userRouter);  
app.use('/api/v1/auth', authRouter);
```

```
app.get('*', (req, res) => {  
  res.sendFile(path.resolve(__dirname, './client/dist', 'index.html'));  
});
```

```
app.use('*', (req, res) => {  
  res.status(404).json({ msg: 'not found' });  
});
```

```
app.use(errorHandlerMiddleware);
```

```
const port = process.env.PORT || 5100;
```

```
try {  
  await mongoose.connect(process.env.MONGO_URL);  
  app.listen(port, () => {  
    console.log(` server running on PORT ${port}...`);  
  });  
} catch (error) {  
  console.log(error);  
  process.exit(1);  
}
```

=====

File: client/index.html

=====

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <link rel="icon" type="image/svg+xml" href="/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Jobify</title>

  </head>

  <body>

    <div id="root"></div>

    <script type="module" src="/src/main.jsx"></script>

  </body>

</html>
```

=====

File: client/package.json

=====

```
{
  "name": "client",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
```

```
"lint": "eslint src --ext js,jsx --report-unused-disable-directives --max-warnings 0",
"preview": "vite preview"
},
"dependencies": {
"@tanstack/react-query": "^4.29.5",
"@tanstack/react-query-devtools": "^4.29.6",
"axios": "^1.3.6",
"dayjs": "^1.11.7",
"react": "^18.2.0",
"react-dom": "^18.2.0",
"react-icons": "^4.8.0",
"react-router-dom": "^6.10.0",
"react-toastify": "^9.1.2",
"recharts": "^2.5.0",
"styled-components": "^5.3.10"
},
"devDependencies": {
"@types/react": "^18.0.37",
"@types/react-dom": "^18.0.11",
"@vitejs/plugin-react": "^4.0.0",
"eslint": "^8.38.0",
"eslint-plugin-react": "^7.32.2",
"eslint-plugin-react-hooks": "^4.6.0",
"eslint-plugin-react-refresh": "^0.3.4",
"vite": "^4.3.9"
}
```



```
}
```

```
=====
```

File: client/vite.config.js

```
=====
```

```
import { defineConfig } from 'vite';
```

```
import react from '@vitejs/plugin-react';
```

```
// https://vitejs.dev/config/
```

```
export default defineConfig({
```

```
  plugins: [react()],
```

```
  server: {
```

```
    proxy: {
```

```
      '/api': {
```

```
        target: 'http://localhost:5100/api',
```

```
        changeOrigin: true,
```

```
        rewrite: (path) => path.replace(/^\/api/, ''),
```

```
      },
```

```
    },
```

```
  },
```

```
});
```

```
=====
```

File: client/.eslintrc.cjs

```
=====

module.exports = {
  env: { browser: true, es2020: true },
  extends: [
    'eslint:recommended',
    'plugin:react/recommended',
    'plugin:react/jsx-runtime',
    'plugin:react-hooks/recommended',
  ],
  parserOptions: { ecmaVersion: 'latest', sourceType: 'module' },
  settings: { react: { version: '18.2' } },
  plugins: ['react-refresh'],
  rules: {
    'react-refresh/only-export-components': 'warn',
  },
}

```

```
=====

File: client/.gitignore
=====

# Logs
logs
*.log
npm-debug.log*
yarn-debug.log*
```

yarn-error.log*
pnpm-debug.log*
lerna-debug.log*
package-lock.json

node_modules
dist
dist-ssr
*.local

Editor directories and files

.vscode/*
!.vscode/extensions.json
.idea
.DS_Store
*.suo
.ntvs
*.njsproj
*.sln
*.sw?

=====

File: client/src/App.jsx

=====

import { RouterProvider, createBrowserRouter } from 'react-router-dom';

```
import { QueryClient, QueryClientProvider } from '@tanstack/react-query';  
import { ReactQueryDevtools } from '@tanstack/react-query-devtools';
```

```
import {  
  HomeLayout,  
  Landing,  
  Register,  
  Login,  
  DashboardLayout,  
  Error,  
  AddJob,  
  Stats,  
  AllJobs,  
  Profile,  
  Admin,  
  EditJob,  
} from './pages';
```

```
import { action as registerAction } from './pages/Register';  
import { action as loginAction } from './pages/Login';  
import { loader as dashboardLoader } from './pages/DashboardLayout';  
import { action as addJobAction } from './pages/AddJob';  
import { loader as allJobsLoader } from './pages/AllJobs';  
import { loader as editJobLoader } from './pages/EditJob';  
import { action as editJobAction } from './pages/EditJob';  
import { action as deleteJobAction } from './pages/DeleteJob';
```

```
import { loader as adminLoader } from './pages/Admin';
import { action as profileAction } from './pages/Profile';
import { loader as statsLoader } from './pages/Stats';
import ErrorElement from './components/ErrorElement';

export const checkDefaultTheme = () => {
  const isDarkTheme = localStorage.getItem('darkTheme') === 'true';
  document.body.classList.toggle('dark-theme', isDarkTheme);
  return isDarkTheme;
};

checkDefaultTheme();

const queryClient = new QueryClient({
  defaultOptions: {
    queries: {
      staleTime: 1000 * 60 * 5,
    },
  },
});

const router = createBrowserRouter([
  {
    path: '/',
    element: <HomeLayout />,
    errorElement: <Error />,
  },
]);
```

```
children: [  
  {  
    index: true,  
    element: <Landing />,  
  },  
  {  
    path: 'register',  
    element: <Register />,  
    action: registerAction,  
  },  
  {  
    path: 'login',  
    element: <Login />,  
    action: loginAction(queryClient),  
  },  
  {  
    path: 'dashboard',  
    element: <DashboardLayout queryClient={queryClient} />,  
    loader: dashboardLoader(queryClient),  
    children: [  
      {  
        index: true,  
        element: <AddJob />,  
        action: addJobAction(queryClient),  
      },  
      {
```

```
path: 'stats',
element: <Stats />,
loader: statsLoader(queryClient),
errorElement: <ErrorElement />,
},
{
path: 'all-jobs',
element: <AllJobs />,
loader: allJobsLoader(queryClient),
errorElement: <ErrorElement />,
},
{
path: 'profile',
element: <Profile />,
action: profileAction(queryClient),
},
{
path: 'admin',
element: <Admin />,
loader: adminLoader,
},
{
path: 'edit-job/:id',
element: <EditJob />,
loader: editJobLoader(queryClient),
action: editJobAction(queryClient),
```

```

    },
    { path: 'delete-job/:id', action: deleteJobAction(queryClient) },
  ],
},
],
},
]);

```

```

const App = () => {
  return (
    <QueryClientProvider client={queryClient}>
      <RouterProvider router={router} />
      <ReactQueryDevtools initialIsOpen={false} />
    </QueryClientProvider>
  );
};

export default App;

```

```

=====

```

File: client/src/index.css

```

=====

```

```

/* ===== GLOBAL CSS ===== */

```

```

*,

```

```

::after,

```



```
::before {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

```
html {  
  font-size: 100%;  
} /*16px*/
```

```
:root {  
  
  /* colors */  
  
  --primary-50: #e0fcff;  
  --primary-100: #bef8fd;  
  --primary-200: #87eaf2;  
  --primary-300: #54d1db;  
  --primary-400: #38bec9;  
  --primary-500: #2cb1bc;  
  --primary-600: #14919b;  
  --primary-700: #0e7c86;  
  --primary-800: #0a6c74;  
  --primary-900: #044e54;
```

```
  /* grey */  
  
  --grey-50: #f8fafc;  
  --grey-100: #f1f5f9;
```

```
--grey-200: #e2e8f0;
--grey-300: #cbd5e1;
--grey-400: #94a3b8;
--grey-500: #64748b;
--grey-600: #475569;
--grey-700: #334155;
--grey-800: #1e293b;
--grey-900: #0f172a;
/* rest of the colors */

--black: #222;
--white: #fff;
--red-light: #f8d7da;
--red-dark: #842029;
--green-light: #d1e7dd;
--green-dark: #0f5132;

--small-text: 0.875rem;
--extra-small-text: 0.7em;
/* rest of the vars */

--border-radius: 0.25rem;
--letter-spacing: 1px;
--transition: 0.3s ease-in-out all;
--max-width: 1120px;
--fixed-width: 600px;
--fluid-width: 90vw;
```

```
--nav-height: 6rem;

/* box shadow*/

--shadow-1: 0 1px 3px 0 rgba(0, 0, 0, 0.1), 0 1px 2px 0 rgba(0, 0, 0, 0.06);

--shadow-2: 0 4px 6px -1px rgba(0, 0, 0, 0.1),
    0 2px 4px -1px rgba(0, 0, 0, 0.06);

--shadow-3: 0 10px 15px -3px rgba(0, 0, 0, 0.1),
    0 4px 6px -2px rgba(0, 0, 0, 0.05);

--shadow-4: 0 20px 25px -5px rgba(0, 0, 0, 0.1),
    0 10px 10px -5px rgba(0, 0, 0, 0.04);

/* DARK MODE */

--dark-mode-bg-color: #333;
--dark-mode-text-color: #f0f0f0;
--dark-mode-bg-secondary-color: #3f3f3f;
--dark-mode-text-secondary-color: var(--grey-300);

--background-color: var(--grey-50);
--text-color: var(--grey-900);
--background-secondary-color: var(--white);
--text-secondary-color: var(--grey-500);
}
```

```
.dark-theme {
  --text-color: var(--dark-mode-text-color);
  --background-color: var(--dark-mode-bg-color);
  --text-secondary-color: var(--dark-mode-text-secondary-color);
}
```

```
--background-secondary-color: var(--dark-mode-bg-secondary-color);  
}  
  
body{  
  background: var(--background-color);  
  color: var(--text-color);  
  font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto,  
    Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;  
  font-weight: 400;  
  line-height: 1;  
}  
p{  
  margin: 0;  
}  
h1,  
h2,  
h3,  
h4,  
h5{  
  margin: 0;  
  font-weight: 400;  
  line-height: 1;  
  text-transform: capitalize;  
  letter-spacing: var(--letter-spacing);  
}
```

```
h1 {  
  font-size: clamp(2rem, 5vw, 5rem); /* Large heading */  
}
```

```
h2 {  
  font-size: clamp(1.5rem, 3vw, 3rem); /* Medium heading */  
}
```

```
h3 {  
  font-size: clamp(1.25rem, 2.5vw, 2.5rem); /* Small heading */  
}
```

```
h4 {  
  font-size: clamp(1rem, 2vw, 2rem); /* Extra small heading */  
}
```

```
h5 {  
  font-size: clamp(0.875rem, 1.5vw, 1.5rem); /* Tiny heading */  
}
```

```
/* BIGGER FONTS */
```

```
/* h1 {  
  font-size: clamp(3rem, 6vw, 6rem);  
}
```

```
h2 {
```

```
font-size: clamp(2.5rem, 5vw, 5rem);  
}
```

```
h3{  
font-size: clamp(2rem, 4vw, 4rem);  
}
```

```
h4{  
font-size: clamp(1.5rem, 3vw, 3rem);  
}
```

```
h5{  
font-size: clamp(1rem, 2vw, 2rem);  
}  
*/
```

```
.text{  
margin-bottom: 1.5rem;  
max-width: 40em;  
}
```

```
small,  
.text-small{  
font-size: var(--small-text);  
}
```

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

```
ul {  
  list-style-type: none;  
  padding: 0;  
}
```

```
.img {  
  width: 100%;  
  display: block;  
  object-fit: cover;  
}
```

```
/* buttons */
```

```
.btn {  
  cursor: pointer;  
  color: var(--white);  
  background: var(--primary-500);  
  border: transparent;  
  border-radius: var(--border-radius);  
  letter-spacing: var(--letter-spacing);  
  padding: 0.375rem 0.75rem;  
  box-shadow: var(--shadow-1);  
  transition: var(--transition);  
  text-transform: capitalize;
```

```
display: inline-block;
}
.btn:hover {
  background: var(--primary-700);
  box-shadow: var(--shadow-3);
}
.btn-hipster {
  color: var(--primary-500);
  background: var(--primary-200);
}
.btn-hipster:hover {
  color: var(--primary-200);
  background: var(--primary-700);
}
.btn-block {
  width: 100%;
}
button:disabled {
  cursor: wait;
}
.danger-btn {
  color: var(--red-dark);
  background: var(--red-light);
}
.danger-btn:hover {
  color: var(--white);
```



```
background: var(--red-dark);
}

/* alerts */

.alert {
padding: 0.375rem 0.75rem;
margin-bottom: 1rem;
border-color: transparent;
border-radius: var(--border-radius);
}

.alert-danger {
color: var(--red-dark);
background: var(--red-light);
}

.alert-success {
color: var(--green-dark);
background: var(--green-light);
}

/* form */

.form {
width: 90vw;
max-width: var(--fixed-width);
background: var(--background-secondary-color);
border-radius: var(--border-radius);
box-shadow: var(--shadow-2);
```

```
padding: 2rem 2.5rem;
margin: 3rem auto;
}
.form-label {
display: block;
font-size: var(--small-text);
margin-bottom: 0.75rem;
text-transform: capitalize;
letter-spacing: var(--letter-spacing);
line-height: 1.5;
}
.form-input,
.form-textarea,
.form-select {
width: 100%;
padding: 0.375rem 0.75rem;
border-radius: var(--border-radius);
background: var(--background-color);
border: 1px solid var(--grey-300);
color: var(--text-color);
}
.form-input,
.form-select,
.form-btn {
height: 35px;
}
```

```
.form-row {  
  margin-bottom: 1rem;  
}
```

```
.form-textarea {  
  height: 7rem;  
}
```

```
::placeholder {  
  font-family: inherit;  
  color: var(--grey-400);  
}
```

```
.form-alert {  
  color: var(--red-dark);  
  letter-spacing: var(--letter-spacing);  
  text-transform: capitalize;  
}
```

```
/* alert */
```

```
@keyframes spinner {  
  to {  
    transform: rotate(360deg);  
  }  
}
```

```
.loading {  
  width: 6rem;
```

```
height: 6rem;

border: 5px solid var(--grey-400);

border-radius: 50%;

border-top-color: var(--primary-500);

animation: spinner 0.6s linear infinite;

}
```

```
/* title */
```

```
.title {

  text-align: center;

}
```

```
.title-underline {

  background: var(--primary-500);

  width: 7rem;

  height: 0.25rem;

  margin: 0 auto;

  margin-top: 1rem;

}
```

```
.container {

  width: var(--fluid-width);

  max-width: var(--max-width);

  margin: 0 auto;

}
```

```
/* BUTTONS AND BADGES */
```

```
.pending {  
  background: #fef3c7;  
  color: #f59e0b;  
}
```

```
.interview {  
  background: #e0e8f9;  
  color: #647acb;  
}
```

```
.declined {  
  background: #ffeeee;  
  color: #d66a6a;  
}
```

```
=====
```

```
File: client/src/main.jsx
```

```
=====
```

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
import App from './App.jsx';  
import 'react-toastify/dist/ReactToastify.css';  
import './index.css';  
import { ToastContainer } from 'react-toastify';
```

```
ReactDOM.createRoot(document.getElementById('root')).render(  
  <>  
    <App />  
    <ToastContainer position='top-center' />  
  </>  
);
```

```
=====
```

File: client/src/assets/css/index.css

```
=====
```

```
/* ===== GLOBAL CSS ===== */
```

```
*,  
::after,  
::before {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

```
html {  
  font-size: 100%;  
} /*16px*/
```

```
:root {
```

/ colors */*

--primary-50: #e0fcff;
--primary-100: #bef8fd;
--primary-200: #87eaf2;
--primary-300: #54d1db;
--primary-400: #38bec9;
--primary-500: #2cb1bc;
--primary-600: #14919b;
--primary-700: #0e7c86;
--primary-800: #0a6c74;
--primary-900: #044e54;

/ grey */*

--grey-50: #f8fafc;
--grey-100: #f1f5f9;
--grey-200: #e2e8f0;
--grey-300: #cbd5e1;
--grey-400: #94a3b8;
--grey-500: #64748b;
--grey-600: #475569;
--grey-700: #334155;
--grey-800: #1e293b;
--grey-900: #0f172a;

/ rest of the colors */*

--black: #222;
--white: #fff;

```
--red-light: #f8d7da;

--red-dark: #842029;

--green-light: #d1e7dd;

--green-dark: #0f5132;


--small-text: 0.875rem;

--extra-small-text: 0.7em;

/* rest of the vars */


--border-radius: 0.25rem;

--letter-spacing: 1px;

--transition: 0.3s ease-in-out all;

--max-width: 1120px;

--fixed-width: 600px;

--fluid-width: 90vw;

--nav-height: 6rem;

/* box shadow*/

--shadow-1: 0 1px 3px 0 rgba(0, 0, 0, 0.1), 0 1px 2px 0 rgba(0, 0, 0, 0.06);

--shadow-2: 0 4px 6px -1px rgba(0, 0, 0, 0.1),

    0 2px 4px -1px rgba(0, 0, 0, 0.06);

--shadow-3: 0 10px 15px -3px rgba(0, 0, 0, 0.1),

    0 4px 6px -2px rgba(0, 0, 0, 0.05);

--shadow-4: 0 20px 25px -5px rgba(0, 0, 0, 0.1),

    0 10px 10px -5px rgba(0, 0, 0, 0.04);

/* DARK MODE */
```



```
--dark-mode-bg-color: #333;  
--dark-mode-text-color: #f0f0f0;  
--dark-mode-bg-secondary-color: #3f3f3f;  
--dark-mode-text-secondary-color: var(--grey-300);
```

```
--background-color: var(--grey-50);  
--text-color: var(--grey-900);  
--background-secondary-color: var(--white);  
--text-secondary-color: var(--grey-500);
```

```
}
```

```
.dark-theme {
```

```
--text-color: var(--dark-mode-text-color);  
--background-color: var(--dark-mode-bg-color);  
--text-secondary-color: var(--dark-mode-text-secondary-color);  
--background-secondary-color: var(--dark-mode-bg-secondary-color);
```

```
}
```

```
body {
```

```
background: var(--background-color);  
color: var(--text-color);  
font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto,  
Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;  
font-weight: 400;  
line-height: 1;
```

```
}
```

```
p{
  margin: 0;
}

h1,
h2,
h3,
h4,
h5{
  margin: 0;
  font-weight: 400;
  line-height: 1;
  text-transform: capitalize;
  letter-spacing: var(--letter-spacing);
}

h1{
  font-size: clamp(2rem, 5vw, 5rem); /* Large heading */
}

h2{
  font-size: clamp(1.5rem, 3vw, 3rem); /* Medium heading */
}

h3{
  font-size: clamp(1.25rem, 2.5vw, 2.5rem); /* Small heading */
}
```

```
h4{  
  font-size: clamp(1rem, 2vw, 2rem); /* Extra small heading */  
}
```

```
h5{  
  font-size: clamp(0.875rem, 1.5vw, 1.5rem); /* Tiny heading */  
}
```

```
/* BIGGER FONTS */
```

```
/* h1 {  
  font-size: clamp(3rem, 6vw, 6rem);  
}
```

```
h2{  
  font-size: clamp(2.5rem, 5vw, 5rem);  
}
```

```
h3{  
  font-size: clamp(2rem, 4vw, 4rem);  
}
```

```
h4{  
  font-size: clamp(1.5rem, 3vw, 3rem);  
}
```

```
h5{  
  font-size: clamp(1rem, 2vw, 2rem);  
}  
*/
```

```
.text{  
  margin-bottom: 1.5rem;  
  max-width: 40em;  
}
```

```
small,  
.text-small{  
  font-size: var(--small-text);  
}
```

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

```
ul{  
  list-style-type: none;  
  padding: 0;  
}
```

```
.img{  
  width: 100%;  
  display: block;
```

```
    object-fit: cover;
}

/* buttons */

.btn {
    cursor: pointer;
    color: var(--white);
    background: var(--primary-500);
    border: transparent;
    border-radius: var(--border-radius);
    letter-spacing: var(--letter-spacing);
    padding: 0.375rem 0.75rem;
    box-shadow: var(--shadow-1);
    transition: var(--transition);
    text-transform: capitalize;
    display: inline-block;
}

.btn:hover {
    background: var(--primary-700);
    box-shadow: var(--shadow-3);
}

.btn-hipster {
    color: var(--primary-500);
    background: var(--primary-200);
}

.btn-hipster:hover {
```

```
    color: var(--primary-200);
    background: var(--primary-700);
}

.btn-block {
    width: 100%;
}

button:disabled {
    cursor: wait;
}

.danger-btn {
    color: var(--red-dark);
    background: var(--red-light);
}

.danger-btn:hover {
    color: var(--white);
    background: var(--red-dark);
}

/* alerts */

.alert {
    padding: 0.375rem 0.75rem;
    margin-bottom: 1rem;
    border-color: transparent;
    border-radius: var(--border-radius);
}

.alert-danger {
```

```
    color: var(--red-dark);

    background: var(--red-light);
}

.alert-success {

    color: var(--green-dark);

    background: var(--green-light);
}

/* form */

.form {

    width: 90vw;

    max-width: var(--fixed-width);

    background: var(--background-secondary-color);

    border-radius: var(--border-radius);

    box-shadow: var(--shadow-2);

    padding: 2rem 2.5rem;

    margin: 3rem auto;
}

.form-label {

    display: block;

    font-size: var(--small-text);

    margin-bottom: 0.75rem;

    text-transform: capitalize;

    letter-spacing: var(--letter-spacing);

    line-height: 1.5;
}
```

```
.form-input,
.form-textarea,
.form-select {
  width: 100%;
  padding: 0.375rem 0.75rem;
  border-radius: var(--border-radius);
  background: var(--background-color);
  border: 1px solid var(--grey-300);
  color: var(--text-color);
}

.form-input,
.form-select,
.form-btn {
  height: 35px;
}

.form-row {
  margin-bottom: 1rem;
}

.form-textarea {
  height: 7rem;
}

::placeholder {
  font-family: inherit;
  color: var(--grey-400);
}
```



```
.form-alert {  
  color: var(--red-dark);  
  letter-spacing: var(--letter-spacing);  
  text-transform: capitalize;  
}  
/* alert */
```

```
@keyframes spinner {  
  to {  
    transform: rotate(360deg);  
  }  
}
```

```
.loading {  
  width: 6rem;  
  height: 6rem;  
  border: 5px solid var(--grey-400);  
  border-radius: 50%;  
  border-top-color: var(--primary-500);  
  animation: spinner 0.6s linear infinite;  
}
```

```
/* title */
```

```
.title {  
  text-align: center;
```

```
}
```

```
.title-underline {  
  background: var(--primary-500);  
  width: 7rem;  
  height: 0.25rem;  
  margin: 0 auto;  
  margin-top: 1rem;  
}
```

```
.container {  
  width: var(--fluid-width);  
  max-width: var(--max-width);  
  margin: 0 auto;  
}
```

```
/* BUTTONS AND BADGES */
```

```
.pending {  
  background: #fef3c7;  
  color: #f59e0b;  
}
```

```
.interview {  
  background: #e0e8f9;  
  color: #647acb;  
}
```

```
.declined {  
  background: #ffeeee;  
  color: #d66a6a;  
}
```

=====

File: client/src/assets/wrappers/BigSidebar.js

=====

```
import styled from 'styled-components';  
  
const Wrapper = styled.aside`  
  display: none;  
  @media (min-width: 992px) {  
    display: block;  
    box-shadow: 1px 0px 0px 0px rgba(0, 0, 0, 0.1);  
    .sidebar-container {  
      background: var(--background-secondary-color);  
      min-height: 100vh;  
      height: 100%;  
      width: 250px;  
      margin-left: -250px;  
      transition: margin-left 0.3s ease-in-out;  
    }  
    .content {  
      position: sticky;
```

```
    top: 0;
}
.show-sidebar {
    margin-left: 0;
}
header {
    height: 6rem;
    display: flex;
    align-items: center;
    padding-left: 2.5rem;
}
.nav-links {
    padding-top: 2rem;
    display: flex;
    flex-direction: column;
}
.nav-link {
    display: flex;
    align-items: center;
    color: var(--text-secondary-color);
    padding: 1rem 0;
    padding-left: 2.5rem;
    text-transform: capitalize;
    transition: padding-left 0.3s ease-in-out;
}
.nav-link:hover {
```

```
padding-left: 3rem;

color: var(--primary-500);

transition: var(--transition);
}

.icon {

font-size: 1.5rem;

margin-right: 1rem;

display: grid;

place-items: center;

}

.active {

color: var(--primary-500);

}

.pending {

background: var(--background-color);

}

}

`;

export default Wrapper;
```

=====

File: client/src/assets/wrappers/ChartsContainer.js

=====

```
import styled from 'styled-components';
```

```
const Wrapper = styled.section`
  margin-top: 4rem;
  text-align: center;
  button {
    background: transparent;
    border-color: transparent;
    text-transform: capitalize;
    color: var(--primary-500);
    font-size: 1.25rem;
    cursor: pointer;
  }
  h4 {
    text-align: center;
    margin-bottom: 0.75rem;
  }
`;
```

```
export default Wrapper;
```

```
=====
```

File: client/src/assets/wrappers/Dashboard.js

```
=====
```

```
import styled from 'styled-components';
```

```
const Wrapper = styled.section`
```

```
.dashboard {
  display: grid;
  grid-template-columns: 1fr;
}

.dashboard-page {
  width: 90vw;
  margin: 0 auto;
  padding: 2rem 0;
}

@media (min-width: 992px) {
  .dashboard {
    grid-template-columns: auto 1fr;
  }

  .dashboard-page {
    width: 90%;
  }
}

`;

export default Wrapper;
```

=====

File: client/src/assets/wrappers/DashboardFormPage.js

=====

```
import styled from 'styled-components';
```

```
const Wrapper = styled.section`  
  border-radius: var(--border-radius);  
  width: 100%;  
  background: var(--background-secondary-color);  
  padding: 3rem 2rem 4rem;  
  .form-title {  
    margin-bottom: 2rem;  
  }  
  .form {  
    margin: 0;  
    border-radius: 0;  
    box-shadow: none;  
    padding: 0;  
    max-width: 100%;  
    width: 100%;  
  }  
  .form-row {  
    margin-bottom: 0;  
  }  
  .form-center {  
    display: grid;  
    row-gap: 1rem;  
  }  
  .form-btn {  
    align-self: end;  
    margin-top: 1rem;
```



```
display: grid;
place-items: center;
}
@media (min-width: 992px) {
  .form-center {
    grid-template-columns: 1fr 1fr;
    align-items: center;
    column-gap: 1rem;
  }
}
@media (min-width: 1120px) {
  .form-center {
    grid-template-columns: 1fr 1fr 1fr;
  }
}
`;
```

```
export default Wrapper;
```

```
=====
```

```
File: client/src/assets/wrappers/ErrorPage.js
```

```
=====
```

```
import styled from 'styled-components';
```

```
const Wrapper = styled.main`
```

```
min-height: 100vh;
text-align: center;
display: flex;
align-items: center;
justify-content: center;
img {
  width: 90vw;
  max-width: 600px;
  display: block;
  margin-bottom: 2rem;
  margin-top: -3rem;
}
h3 {
  margin-bottom: 0.5rem;
}
p {
  line-height: 1.5;
  margin-top: 0.5rem;
  margin-bottom: 1rem;
  color: var(--text-secondary-color);
}
a {
  color: var(--primary-500);
  text-transform: capitalize;
}
`;
```

```
export default Wrapper;
```

```
=====
```

```
File: client/src/assets/wrappers/Job.js
```

```
=====
```

```
import styled from 'styled-components';
```

```
const Wrapper = styled.article`
```

```
  background: var(--background-secondary-color);
```

```
  border-radius: var(--border-radius);
```

```
  display: grid;
```

```
  grid-template-rows: 1fr auto;
```

```
  box-shadow: var(--shadow-2);
```

```
  header {
```

```
    padding: 1rem 1.5rem;
```

```
    border-bottom: 1px solid var(--grey-100);
```

```
    display: grid;
```

```
    grid-template-columns: auto 1fr;
```

```
    align-items: center;
```

```
  }
```

```
  .main-icon {
```

```
    width: 60px;
```

```
    height: 60px;
```

```
    display: grid;
```

```
place-items: center;
background: var(--primary-500);
border-radius: var(--border-radius);
font-size: 1.5rem;
font-weight: 700;
text-transform: uppercase;
color: var(--white);
margin-right: 2rem;
}
.info {
  h5 {
    margin-bottom: 0.5rem;
  }
  p {
    margin: 0;
    text-transform: capitalize;
    letter-spacing: var(--letter-spacing);
    color: var(--text-secondary-color);
  }
}
.content {
  padding: 1rem 1.5rem;
}
.content-center {
  display: grid;
  margin-top: 1rem;
```

```
margin-bottom: 1.5rem;
grid-template-columns: 1fr;
row-gap: 1.5rem;
align-items: center;
@media (min-width: 576px) {
  grid-template-columns: 1fr 1fr;
}
}

.status {
  border-radius: var(--border-radius);
  text-transform: capitalize;
  letter-spacing: var(--letter-spacing);
  text-align: center;
  width: 100px;
  height: 30px;
  display: grid;
  align-items: center;
}

.actions {
  margin-top: 1rem;
  display: flex;
  align-items: center;
}

.edit-btn,
.delete-btn {
  height: 30px;
```

```
    font-size: 0.85rem;

    display: flex;

    align-items: center;
  }

  .edit-btn {

    margin-right: 0.5rem;
  }
`;
```

```
export default Wrapper;
```

```
=====
```

```
File: client/src/assets/wrappers/JobInfo.js
```

```
=====
```

```
import styled from 'styled-components';
```

```
const Wrapper = styled.div`
```

```
  display: flex;
```

```
  align-items: center;
```

```
  .job-icon {
```

```
    font-size: 1rem;
```

```
    margin-right: 1rem;
```

```
    display: flex;
```

```
    align-items: center;
```

```
    svg {
```

```
    color: var(--text-secondary-color);
  }
}
.job-text {
  text-transform: capitalize;
  letter-spacing: var(--letter-spacing);
}
`;
export default Wrapper;
```

=====

File: client/src/assets/wrappers/JobsContainer.js

=====

```
import styled from 'styled-components';
```

```
const Wrapper = styled.section`
```

```
  margin-top: 4rem;
```

```
  h2 {
```

```
    text-transform: none;
```

```
  }
```

```
  & > h5 {
```

```
    font-weight: 700;
```

```
    margin-bottom: 1.5rem;
```

```
  }
```

```
  .jobs {
```

```
display: grid;
grid-template-columns: 1fr;
row-gap: 2rem;
}
@media (min-width: 1120px) {
  .jobs {
    grid-template-columns: 1fr 1fr;
    gap: 2rem;
  }
}
`;
export default Wrapper;
```

=====

File: client/src/assets/wrappers/LandingPage.js

=====

```
import styled from 'styled-components';
```

```
const Wrapper = styled.section`
```

```
  nav {
```

```
    width: var(--fluid-width);
```

```
    max-width: var(--max-width);
```

```
    margin: 0 auto;
```

```
    height: var(--nav-height);
```

```
    display: flex;
```



```
    align-items: center;
}
.page {
    min-height: calc(100vh - var(--nav-height));
    display: grid;
    align-items: center;
    margin-top: -3rem;
}
h1 {
    font-weight: 700;
    span {
        color: var(--primary-500);
    }
    margin-bottom: 1.5rem;
}
p {
    line-height: 2;
    color: var(--text-secondary-color);
    margin-bottom: 1.5rem;
    max-width: 35em;
}
.register-link {
    margin-right: 1rem;
}
.main-img {
    display: none;
```

```
}  
.btn {  
  padding: 0.75rem 1rem;  
}  
@media (min-width: 992px) {  
  .page {  
    grid-template-columns: 1fr 400px;  
    column-gap: 3rem;  
  }  
  .main-img {  
    display: block;  
  }  
}  
`;  
export default Wrapper;
```

=====

File: client/src/assets/wrappers/LogoutContainer.js

=====

```
import styled from 'styled-components';
```

```
const Wrapper = styled.div`
```

```
  position: relative;
```

```
  .logout-btn {
```

```
    display: flex;
```

```
    align-items: center;
    justify-content: center;
    gap: 0 0.5rem;
}

.img{
    width: 25px;
    height: 25px;
    border-radius: 50%;
}

.dropdown {
    position: absolute;
    top: 45px;
    left: 0;
    width: 100%;
    box-shadow: var(--shadow-2);
    text-align: center;
    visibility: hidden;
    border-radius: var(--border-radius);
    background: var(--primary-500);
}

.show-dropdown {
    visibility: visible;
}

.dropdown-btn {
    border-radius: var(--border-radius);
    padding: 0.5rem;
```

```
background: transparent;
border-color: transparent;
color: var(--white);
letter-spacing: var(--letter-spacing);
text-transform: capitalize;
cursor: pointer;
width: 100%;
height: 100%;
}
`;
```

```
export default Wrapper;
```

```
=====
```

```
File: client/src/assets/wrappers/Navbar.js
```

```
=====
```

```
import styled from 'styled-components';
```

```
const Wrapper = styled.nav`
  height: var(--nav-height);
  display: flex;
  align-items: center;
  justify-content: center;
  box-shadow: 0 1px 0 0 rgba(0, 0, 0, 0.1);
  background: var(--background-secondary-color);
```

```
.nav-center {  
  display: flex;  
  width: 90vw;  
  align-items: center;  
  justify-content: space-between;  
}
```

```
.toggle-btn {  
  background: transparent;  
  border-color: transparent;  
  font-size: 1.75rem;  
  color: var(--primary-500);  
  cursor: pointer;  
  display: flex;  
  align-items: center;  
}
```

```
.logo-text {  
  display: none;  
}
```

```
.logo {  
  display: flex;  
  align-items: center;  
  width: 100px;  
}
```

```
.btn-container {  
  display: flex;  
  align-items: center;
```

```
}  
  
@media (min-width: 992px) {  
  position: sticky;  
  top: 0;  
  .nav-center {  
    width: 90%;  
  }  
  .logo {  
    display: none;  
  }  
  .logo-text {  
    display: block;  
  }  
}  
`;  
  
export default Wrapper;
```

=====

File: client/src/assets/wrappers/PageBtnContainer.js

=====

```
import styled from 'styled-components';
```

```
const Wrapper = styled.section`
```

```
  height: 6rem;
```

```
  margin-top: 2rem;
```

```
display: flex;
align-items: center;
justify-content: end;
flex-wrap: wrap;
gap: 1rem;
.btn-container {
  background: var(--background-secondary-color);
  border-radius: var(--border-radius);
  display: flex;
}
.page-btn {
  background: transparent;
  border-color: transparent;
  width: 50px;
  height: 40px;
  font-weight: 700;
  font-size: 1.25rem;
  color: var(--primary-500);
  border-radius: var(--border-radius);
  cursor: pointer;
}
.active{
  background: var(--primary-500);
  color: var(--white);
}
```

```
.prev-btn,.next-btn{
  background: var(--background-secondary-color);
  border-color: transparent;
  border-radius: var(--border-radius);

  width: 100px;
  height: 40px;
  color: var(--primary-500);
  text-transform:capitalize;
  letter-spacing:var(--letter-spacing);
  display:flex;
  align-items:center;
  justify-content:center;
  gap:0.5rem;
  cursor:pointer;
}

.prev-btn:hover,.next-btn:hover{
  background:var(--primary-500);
  color: var(--white);
  transition:var(--transition);
}

.dots{
  display:grid;
  place-items:center;
  cursor:text;
}
```



```
`;
```

```
export default Wrapper;
```

```
=====
```

```
File: client/src/assets/wrappers/RegisterAndLoginPage.js
```

```
=====
```

```
import styled from 'styled-components';
```

```
const Wrapper = styled.section`
```

```
  min-height: 100vh;
```

```
  display: grid;
```

```
  align-items: center;
```

```
  .logo {
```

```
    display: block;
```

```
    margin: 0 auto;
```

```
    margin-bottom: 1.38rem;
```

```
  }
```

```
  .form {
```

```
    max-width: 400px;
```

```
    border-top: 5px solid var(--primary-500);
```

```
  }
```

```
  h4 {
```

```
    text-align: center;
```

```
    margin-bottom: 1.38rem;
```

```
  }
```

```
p {
  margin-top: 1rem;
  text-align: center;
  line-height: 1.5;
}

.btn {
  margin-top: 1rem;
}

.member-btn {
  color: var(--primary-500);
  letter-spacing: var(--letter-spacing);
  margin-left: 0.25rem;
}

`;

export default Wrapper;
```

=====

File: client/src/assets/wrappers/SmallSidebar.js

=====

```
import styled from 'styled-components';
```

```
const Wrapper = styled.aside`
  @media (min-width: 992px) {
    display: none;
  }
}
```

```
.sidebar-container {  
  position: fixed;  
  inset: 0;  
  background: rgba(0, 0, 0, 0.7);  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  z-index: -1;  
  opacity: 0;  
  transition: var(--transition);  
  visibility: hidden;  
}  
  
.show-sidebar {  
  z-index: 99;  
  opacity: 1;  
  visibility: visible;  
}  
  
.content {  
  background: var(--background-secondary-color);  
  width: var(--fluid-width);  
  height: 95vh;  
  border-radius: var(--border-radius);  
  padding: 4rem 2rem;  
  position: relative;  
  display: flex;  
  align-items: center;
```

```
    flex-direction: column;
}

.close-btn {
    position: absolute;
    top: 10px;
    left: 10px;
    background: transparent;
    border-color: transparent;
    font-size: 2rem;
    color: var(--red-dark);
    cursor: pointer;
}

.nav-links {
    padding-top: 2rem;
    display: flex;
    flex-direction: column;
}

.nav-link {
    display: flex;
    align-items: center;
    color: var(--text-secondary-color);
    padding: 1rem 0;
    text-transform: capitalize;
    transition: var(--transition);
}

.nav-link:hover {
```

```
    color: var(--primary-500);
  }
  .icon {
    font-size: 1.5rem;
    margin-right: 1rem;
    display: grid;
    place-items: center;
  }
  .active {
    color: var(--primary-500);
  }
`;
export default Wrapper;
```

=====

File: client/src/assets/wrappers/StatItem.js

=====

```
import styled from 'styled-components';

const Wrapper = styled.article`
  padding: 2rem;
  background: var(--background-secondary-color);
  border-bottom: 5px solid ${props => props.color};
  border-radius: var(--border-radius);
```

```
header {  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
}  
  
.count {  
  display: block;  
  font-weight: 700;  
  font-size: 50px;  
  color: ${props => props.color};  
  line-height: 2;  
}  
  
.title {  
  margin: 0;  
  text-transform: capitalize;  
  letter-spacing: var(--letter-spacing);  
  text-align: left;  
  margin-top: 0.5rem;  
  font-size: 1.25rem;  
}  
  
.icon {  
  width: 70px;  
  height: 60px;  
  background: ${props => props.bcg};  
  border-radius: var(--border-radius);  
  display: flex;
```

```
    align-items: center;
    justify-content: center;
    svg{
      font-size: 2rem;
      color: ${props => props.color};
    }
  }
`;
```

```
export default Wrapper;
```

```
=====
```

```
File: client/src/assets/wrappers/StatsContainer.js
```

```
=====
```

```
import styled from 'styled-components';
```

```
const Wrapper = styled.section`
```

```
  display: grid;
```

```
  row-gap: 2rem;
```

```
  @media (min-width: 768px) {
```

```
    grid-template-columns: 1fr 1fr;
```

```
    column-gap: 1rem;
```

```
  }
```

```
  @media (min-width: 1120px) {
```

```
    grid-template-columns: 1fr 1fr 1fr;
```

```
}  
`;  
  
export default Wrapper;
```

=====

File: client/src/assets/wrappers/Testing.js

=====

```
import styled from 'styled-components'
```

```
const Wrapper = styled.main`  
  nav {  
    width: var(--fluid-width);  
    max-width: var(--max-width);  
    margin: 0 auto;  
    height: var(--nav-height);  
    display: flex;  
    align-items: center;  
  }  
  .page {  
    min-height: calc(100vh - var(--nav-height));  
    display: grid;  
    align-items: center;  
    margin-top: -3rem;  
  }  
  h1 {
```



```
font-weight: 700;

span {
  color: var(--primary-500);
}
}

p {
  color: var(--grey-600);
}

.main-img {
  display: none;
}

@media (min-width: 992px) {
  .page {
    grid-template-columns: 1fr 1fr;
    column-gap: 3rem;
  }
  .main-img {
    display: block;
  }
}
```

export default Wrapper

=====

File: client/src/assets/wrappers/ThemeToggle.js

=====

```
import styled from 'styled-components';
```

```
const Wrapper = styled.button`
```

```
  background: transparent;
```

```
  border-color: transparent;
```

```
  width: 3.5rem;
```

```
  height: 2rem;
```

```
  display: grid;
```

```
  place-items: center;
```

```
  cursor: pointer;
```

```
  .toggle-icon {
```

```
    font-size: 1.15rem;
```

```
    color: var(--text-color);
```

```
  }
```

```
`;
```

```
export default Wrapper;
```

=====

File: client/src/components/AreaChart.jsx

=====

```
import {
```

```
  ResponsiveContainer,
```

```
  AreaChart,
```

```
  Area,
```

```
XAxis,  
YAxis,  
CartesianGrid,  
Tooltip,  
} from 'recharts';
```

```
const AreaChartComponent = ({ data }) => {  
  return (  
    <ResponsiveContainer width='100%' height={300}>  
      <AreaChart data={data} margin={{ top: 50 }}>  
        <CartesianGrid strokeDasharray='3 3' />  
        <XAxis dataKey='date' />  
        <YAxis allowDecimals={false} />  
        <Tooltip />  
        <Area type='monotone' dataKey='count' stroke='#2cb1bc' fill='#bef8fd' />  
      </AreaChart>  
    </ResponsiveContainer>  
  );  
};  
  
export default AreaChartComponent;
```

```
=====
```

File: client/src/components/BarChart.jsx

```
=====
```

```
import {
```

```
BarChart,  
Bar,  
XAxis,  
YAxis,  
CartesianGrid,  
Tooltip,  
ResponsiveContainer,  
} from 'recharts';
```

```
const BarChartComponent = ({ data }) => {  
  return (  
    <ResponsiveContainer width='100%' height={300}>  
      <BarChart data={data} margin={{ top: 50 }}>  
        <CartesianGrid strokeDasharray='3 3' />  
        <XAxis dataKey='date' />  
        <YAxis allowDecimals={false} />  
        <Tooltip />  
        <Bar dataKey='count' fill='#2cb1bc' barSize={75} />  
      </BarChart>  
    </ResponsiveContainer>  
  );  
};  
  
export default BarChartComponent;
```

```
=====
```

File: client/src/components/BigSidebar.jsx

=====

```
import Wrapper from '../assets/wrappers/BigSidebar';
import NavLinks from './NavLinks';
import Logo from './Logo';
import { useDashboardContext } from '../pages/DashboardLayout';

const BigSidebar = () => {
  const { showSidebar } = useDashboardContext();

  return (
    <Wrapper>
      <div
        className={
          showSidebar ? 'sidebar-container' : 'sidebar-container show-sidebar'
        }
      >
        <div className='content'>
          <header>
            <Logo />
          </header>
          <NavLinks isBigSidebar />
        </div>
      </div>
    </Wrapper>
  );
};
```

```
export default BigSidebar;
```

```
=====
```

```
File: client/src/components/ChartsContainer.jsx
```

```
=====
```

```
import { useState } from 'react';
```

```
import BarChart from './BarChart';
```

```
import AreaChart from './AreaChart';
```

```
import Wrapper from '../assets/wrappers/ChartsContainer';
```

```
const ChartsContainer = ({ data }) => {
```

```
  const [barChart, setBarChart] = useState(true);
```

```
  return (
```

```
    <Wrapper>
```

```
      <h4>Monthly Applications</h4>
```

```
      <button type='button' onClick={() => setBarChart(!barChart)}>
```

```
        {barChart ? 'Area Chart' : 'Bar Chart'}
```

```
      </button>
```

```
      {barChart ? <BarChart data={data} /> : <AreaChart data={data} />}
```

```
    </Wrapper>
```

```
  );
```

```
};
```

```
export default ChartsContainer;
```

=====

File: client/src/components/ErrorMessage.jsx

=====

```
import { useRouteError } from 'react-router-dom';
```

```
const ErrorMessage = () => {  
  const error = useRouteError();  
  console.log(error);  
  return <h4>There was an error...</h4>;  
};  
export default ErrorMessage;
```

=====

File: client/src/components/FormRow.jsx

=====

```
const FormRow = ({ type, name, labelText, defaultValue, onChange }) => {  
  return (  
    <div className='form-row'>  
      <label htmlFor={name} className='form-label'>  
        {labelText || name}  
      </label>  
      <input  
        type={type}  
        id={name}
```

```

      name={name}
      className='form-input'
      defaultValue={defaultValue || ''}
      onChange={onChange}
      required
    />
  </div>

);
};

export default FormRow;

```

```
=====
```

File: client/src/components/FormRowSelect.jsx

```
=====
```

```

const FormRowSelect = ({
  name,
  labelText,
  list,
  defaultValue = '',
  onChange,
}) => {
  return (
    <div className='form-row'>
      <label htmlFor={name} className='form-label'>
        {labelText || name}

```



```

</label>

<select
  name={name}
  id={name}
  className='form-select'
  defaultValue={defaultValue}
  onChange={onChange}
>
  {list.map((itemValue) => {
    return (
      <option key={itemValue} value={itemValue}>
        {itemValue}
      </option>
    );
  })}
</select>
</div>

);

};

export default FormRowSelect;

```

```

=====

File: client/src/components/Job.jsx

=====

```

```

import { FaLocationArrow, FaBriefcase, FaCalendarAlt } from 'react-icons/fa';

```

```
import { Link, Form } from 'react-router-dom';  
import Wrapper from '../assets/wrappers/Job';  
import JobInfo from './JobInfo';  
import day from 'dayjs';  
import advancedFormat from 'dayjs/plugin/advancedFormat';  
day.extend(advancedFormat);
```

```
const Job = ({  
  _id,  
  position,  
  company,  
  jobLocation,  
  jobType,  
  createdAt,  
  jobStatus,  
}) => {  
  const date = day(createdAt).format('MMM Do, YYYY');  
  return (  
    <Wrapper>  
      <header>  
        <div className='main-icon'>{company.charAt(0)}</div>  
        <div className='info'>  
          <h5>{position}</h5>  
          <p>{company}</p>  
        </div>  
      </header>
```

```

<div className='content'>
  <div className='content-center'>
    <JobInfo icon={<FaLocationArrow />} text={jobLocation} />
    <JobInfo icon={<FaCalendarAlt />} text={date} />
    <JobInfo icon={<FaBriefcase />} text={jobType} />
    <div className={` status ${jobStatus}`}>{jobStatus}</div>
  </div>
  <footer className='actions'>
    <Link to={` ../edit-job/${_id}`} className='btn edit-btn'>
      Edit
    </Link>
    <Form method='post' action={` ../delete-job/${_id}`}>
      <button type='submit' className='btn delete-btn'>
        Delete
      </button>
    </Form>
  </footer>
</div>
</Wrapper>
);
};
export default Job;

```

=====

File: client/src/components/JobInfo.jsx

=====

```
import Wrapper from '../assets/wrappers/JobInfo';
```

```
const JobInfo = ({ icon, text }) => {  
  return (  
    <Wrapper>  
      <span className='job-icon'>{icon}</span>  
      <span className='job-text'>{text}</span>  
    </Wrapper>  
  );  
};  
export default JobInfo;
```

=====

File: client/src/components/JobsContainer.jsx

=====

```
import Job from './Job';  
import Wrapper from '../assets/wrappers/JobsContainer';  
import { useAllJobsContext } from '../pages/AllJobs';  
import PageBtnContainer from './PageBtnContainer';  
const JobsContainer = () => {  
  const { data } = useAllJobsContext();  
  
  const { jobs, totalJobs, numOfPages } = data;  
  if (jobs.length === 0) {
```

```
return (  
  <Wrapper>  
    <h2>No jobs to display...</h2>  
  </Wrapper>  
);  
}  
return (  
  <Wrapper>  
    <h5>  
      {totalJobs} job{jobs.length > 1 && 's'} found  
    </h5>  
    <div className='jobs'>  
      {jobs.map((job) => {  
        return <Job key={job._id} {...job} />;  
      })}  
    </div>  
    {numOfPages > 1 && <PageBtnContainer />}  
  </Wrapper>  
);  
};  
export default JobsContainer;
```

```
=====
```

File: client/src/components/Loading.jsx

```
=====
```

```
const Loading = () => {  
  return <div className='loading'></div>;  
};  
export default Loading;
```

=====

File: client/src/components/Logo.jsx

=====

```
import logo from '../assets/images/logo.svg';  
  
const Logo = () => {  
  return <img src={logo} alt='jobify' className='logo' />;  
};  
  
export default Logo;
```

=====

File: client/src/components/LogoutContainer.jsx

=====

```
import { FaUserCircle, FaCaretDown } from 'react-icons/fa';  
import Wrapper from '../assets/wrappers/LogoutContainer';  
import { useState } from 'react';  
import { useDashboardContext } from '../pages/DashboardLayout';
```

```
const LogoutContainer = () => {  
  const [showLogout, setShowLogout] = useState(false);  
  const { user, logoutUser } = useDashboardContext();  
  
  return (  
    <Wrapper>  
      <button  
        type='button'  
        className='btn logout-btn'  
        onClick={() => setShowLogout(!showLogout)}  
      >  
        {user.avatar ? (  
          <img src={user.avatar} alt='avатар' className='img' />  
        ) : (  
          <FaUserCircle />  
        )}  
        {user?.name}  
        <FaCaretDown />  
      </button>  
      <div className={showLogout ? 'dropdown show-dropdown' : 'dropdown'}>  
        <button type='button' className='dropdown-btn' onClick={logoutUser}>  
          logout  
        </button>  
      </div>  
    </Wrapper>  
  );
```

```
};
```

```
export default LogoutContainer;
```

```
=====
```

```
File: client/src/components/NavLinks.jsx
```

```
=====
```

```
import { useDashboardContext } from '../pages/DashboardLayout';
```

```
import links from '../utils/links';
```

```
import { NavLink } from 'react-router-dom';
```

```
const NavLinks = ({ isBigSidebar }) => {
```

```
  const { toggleSidebar, user } = useDashboardContext();
```

```
  return (
```

```
    <div className='nav-links'>
```

```
      {links.map((link) => {
```

```
        const { text, path, icon } = link;
```

```
        const { role } = user;
```

```
        if (path === 'admin' && role !== 'admin') return;
```

```
        return (
```

```
          <NavLink
```

```
            to={path}
```

```
            key={text}
```

```
            className='nav-link'
```

```
            onClick={isBigSidebar ? null : toggleSidebar}
```

```
          )
```



```

    >
    <span className='icon'>{icon}</span>
    {text}
  </NavLink>
);
}}
</div>
);
};
export default NavLinks;

```

```

=====
File: client/src/components/Navbar.jsx
=====
import Wrapper from '../assets/wrappers/Navbar';
import { FaAlignLeft } from 'react-icons/fa';
import Logo from './Logo';
import { useDashboardContext } from '../pages/DashboardLayout';
import LogoutContainer from './LogoutContainer';
import ThemeToggle from './ThemeToggle';
const Navbar = () => {
  const { toggleSidebar } = useDashboardContext();
  return (
    <Wrapper>
      <div className='nav-center'>

```

```
<button type='button' className='toggle-btn' onClick={toggleSidebar}>
  <FaAlignLeft />
</button>

<div>
  <Logo />
  <h4 className='logo-text'>dashboard</h4>
</div>

<div className='btn-container'>
  <ThemeToggle />
  <LogoutContainer />
</div>
</div>
</Wrapper>

);
};

export default Navbar;
```

=====

File: client/src/components/PageBtnContainer.jsx

=====

```
import { HiChevronDoubleLeft, HiChevronDoubleRight } from 'react-icons/hi';
import Wrapper from '../assets/wrappers/PageBtnContainer';
import { useLocation, Link, useNavigate } from 'react-router-dom';
import { useAllJobsContext } from '../pages/AllJobs';
```

```

const PageBtnContainer = () => {
  const {
    data: { numOfPages, currentPage },
  } = useAllJobsContext();
  const pages = Array.from({ length: numOfPages }, (_, index) => {
    return index + 1;
  });

  const { search, pathname } = useLocation();
  const navigate = useNavigate();

  const handlePageChange = (pageNumber) => {
    const searchParams = new URLSearchParams(search);
    searchParams.set('page', pageNumber);
    navigate(` ${pathname}?${searchParams.toString()}` );
  };

  const addPageButton = ({ pageNumber, activeClass }) => {
    return (
      <button
        className={` btn page-btn ${activeClass} && 'active'`}
        key={pageNumber}
        onClick={() => handlePageChange(pageNumber)}
      >
        {pageNumber}
      </button>
    )
  }
}

```

```
);
```

```
};
```

```
const renderPageButtons = () => {
```

```
  const pageButtons = [];
```

```
  // first page
```

```
  pageButtons.push(
```

```
    addPageButton({ pageNumber: 1, activeClass: currentPage === 1 })
```

```
  );
```

```
  // dots
```

```
  if (currentPage > 3) {
```

```
    pageButtons.push(
```

```
      <span className='page-btn dots' key='dots-1'>
```

```
        ...
```

```
      </span>
```

```
    );
```

```
  }
```

```
  // one before current page
```

```
  if (currentPage !== 1 && currentPage !== 2) {
```

```
    pageButtons.push(
```

```
      addPageButton({
```

```
        pageNumber: currentPage - 1,
```

```
        activeClass: false,
```

```
      })
```

```
    );
```

```

}

// current page
if (currentPage !== 1 && currentPage !== numOfPages) {
  pageButtons.push(
    addPageButton({
      pageNumber: currentPage,
      activeClass: true,
    })
  );
}

// one after current page

if (currentPage !== numOfPages && currentPage !== numOfPages - 1) {
  pageButtons.push(
    addPageButton({
      pageNumber: currentPage + 1,
      activeClass: false,
    })
  );
}

if (currentPage < numOfPages - 2) {
  pageButtons.push(
    <span className='page-btn dots' key='dots+1'>
      ...
    </span>
  );
}

```

```

}

pageButtons.push(
  addPageButton({
    pageNumber: numOfPages,
    activeClass: currentPage === numOfPages,
  })
);

return pageButtons;
};

return (
  <Wrapper>
    <button
      className='btn prev-btn'
      onClick={() => {
        let prevPage = currentPage - 1;
        if (prevPage < 1) prevPage = numOfPages;
        handlePageChange(prevPage);
      }}
    >
      <HiChevronDoubleLeft />
      prev
    </button>
    <div className='btn-container'>{renderPageButtons()}</div>
    <button
      className='btn next-btn'

```

```

onClick={() => {
  let nextPage = currentPage + 1;
  if (nextPage > numOfPages) nextPage = 1;
  handlePageChange(nextPage);
}}
>

next

<HiChevronDoubleRight />

</button>

</Wrapper>

);
};

export default PageBtnContainer;

```

=====

File: client/src/components/SearchContainer.jsx

=====

```

import { FormRow, FormRowSelect, SubmitBtn } from '!';
import Wrapper from '../assets/wrappers/DashboardFormPage';
import { Form, useSubmit, Link } from 'react-router-dom';
import { JOB_TYPE, JOB_STATUS, JOB_SORT_BY } from '../../utils/constants';
import { useAllJobsContext } from '../pages/AllJobs';

const SearchContainer = () => {
  const { searchValues } = useAllJobsContext();

```

```
const { search, jobStatus, jobType, sort } = searchValues;

const submit = useSubmit();
```

```
const debounce = (onChange) => {
  let timeout;
  return (e) => {
    const form = e.currentTarget.form;
    clearTimeout(timeout);
    timeout = setTimeout(() => {
      onChange(form);
    }, 2000);
  };
};

return (
  <Wrapper>
    <Form className='form'>
      <h5 className='form-title'>search form</h5>
      <div className='form-center'>
        <FormRow
          type='search'
          name='search'
          defaultValue={search}
          onChange={debounce((form) => {
            submit(form);
          })}
        />
      </div>
    </Form>
  </Wrapper>
);
```



```
<FormRowSelect
  labelText='job status'
  name='jobStatus'
  list={['all', ...Object.values(JOB_STATUS)]]
  defaultValue={jobStatus}
  onChange={(e) => {
    submit(e.currentTarget.form);
  }}
/>
```

```
<FormRowSelect
  labelText='job type'
  name='jobType'
  list={['all', ...Object.values(JOB_TYPE)]]
  defaultValue={jobType}
  onChange={(e) => {
    submit(e.currentTarget.form);
  }}
/>
```

```
<FormRowSelect
  name='sort'
  defaultValue={sort}
  list={['...Object.values(JOB_SORT_BY)]]
  onChange={(e) => {
    submit(e.currentTarget.form);
  }}
```

```

    />

    <Link to='/dashboard/all-jobs' className='btn form-btn delete-btn'>

      Reset Search Values

    </Link>

  </div>

</Form>

</Wrapper>

);

};

export default SearchContainer;

```

```

=====

File: client/src/components/SmallSidebar.jsx

=====

import { FaTimes } from 'react-icons/fa';

import Wrapper from '../assets/wrappers/SmallSidebar';

import { useDashboardContext } from '../pages/DashboardLayout';

import Logo from './Logo';

import NavLinks from './NavLinks';

const SmallSidebar = () => {

  const { showSidebar, toggleSidebar } = useDashboardContext();

  return (

    <Wrapper>

```

```

<div
  className={
    showSidebar ? 'sidebar-container show-sidebar' : 'sidebar-container'
  }
>
  <div className='content'>
    <button type='button' className='close-btn' onClick={toggleSidebar}>
      <FaTimes />
    </button>
    <header>
      <Logo />
    </header>
    <NavLinks />
  </div>
</div>
</Wrapper>
);
};

export default SmallSidebar;

```

```
=====
```

File: client/src/components/StatItem.jsx

```
=====
```

```
import Wrapper from '../assets/wrappers/StatItem';
```

```

const StatItem = ({ count, title, icon, color, bcg }) => {
  return (
    <Wrapper color={color} bcg={bcg}>
      <header>
        <span className='count'>{count}</span>
        <span className='icon'>{icon}</span>
      </header>
      <h5 className='title'>{title}</h5>
    </Wrapper>
  );
};

export default StatItem;

```

=====

File: client/src/components/StatsContainer.jsx

=====

```

import { FaSuitcaseRolling, FaCalendarCheck, FaBug } from 'react-icons/fa';
import Wrapper from '../assets/wrappers/StatsContainer';
import StatItem from './StatItem';

const StatsContainer = ({ defaultStats }) => {
  const stats = [
    {
      title: 'pending applications',
      count: defaultStats?.pending || 0,
      icon: <FaSuitcaseRolling />,

```

```

    color: '#f59e0b',
    bcg: '#fef3c7',
  },
  {
    title: 'interviews scheduled',
    count: defaultStats?.interview || 0,
    icon: <FaCalendarCheck />,
    color: '#647acb',
    bcg: '#e0e8f9',
  },
  {
    title: 'jobs declined',
    count: defaultStats?.declined || 0,
    icon: <FaBug />,
    color: '#d66a6a',
    bcg: '#ffeeee',
  },
];
return (
  <Wrapper>
    {stats.map((item) => {
      return <StatItem key={item.title} {...item} />;
    })}
  </Wrapper>
);
};

```

```
export default StatsContainer;
```

```
=====
```

```
File: client/src/components/SubmitBtn.jsx
```

```
=====
```

```
import { useNavigation } from 'react-router-dom';

const SubmitBtn = ({ formBtn }) => {
  const navigation = useNavigation();
  const isSubmitting = navigation.state === 'submitting';
  return (
    <button
      type='submit'
      className={` btn btn-block ${formBtn && 'form-btn'} `}
      disabled={isSubmitting}
    >
      {isSubmitting ? 'submitting' : 'submit'}
    </button>
  );
};

export default SubmitBtn;
```

```
=====
```

```
File: client/src/components/ThemeToggle.jsx
```

```
=====
```

```

import { BsFillSunFill, BsFillMoonFill } from 'react-icons/bs';
import Wrapper from '../assets/wrappers/ThemeToggle';
import { useDashboardContext } from '../pages/DashboardLayout';

const ThemeToggle = () => {
  const { isDarkTheme, toggleDarkTheme } = useDashboardContext();
  return (
    <Wrapper onClick={toggleDarkTheme}>
      {isDarkTheme ? (
        <BsFillSunFill className='toggle-icon' />
      ) : (
        <BsFillMoonFill />
      )}
    </Wrapper>
  );
};

export default ThemeToggle;

```

=====

File: client/src/components/index.js

=====

```

export { default as Logo } from './Logo';
export { default as FormRow } from './FormRow';
export { default as BigSidebar } from './BigSidebar';
export { default as SmallSidebar } from './SmallSidebar';

```

```
export { default as Navbar } from './Navbar';
export { default as FormRowSelect } from './FormRowSelect';
export { default as JobsContainer } from './JobsContainer';
export { default as SearchContainer } from './SearchContainer';
export { default as StatItem } from './StatItem';
export { default as SubmitBtn } from './SubmitBtn';
export { default as ChartsContainer } from './ChartsContainer';
export { default as StatsContainer } from './StatsContainer';
export { default as Loading } from './Loading';
```

=====

File: client/src/pages/AddJob.jsx

=====

```
import { FormRow, FormRowSelect, SubmitBtn } from '../components';
import Wrapper from '../assets/wrappers/DashboardFormPage';
import { useOutletContext } from 'react-router-dom';
import { JOB_STATUS, JOB_TYPE } from '../../utils/constants';
import { Form, redirect } from 'react-router-dom';
import { toast } from 'react-toastify';
import customFetch from '../utils/customFetch';
```

```
export const action =
  (queryClient) =>
  async ({ request }) => {
    const formData = await request.formData();
```



```

const data = Object.fromEntries(formData);

try {
  await customFetch.post('/jobs', data);
  queryClient.invalidateQueries(['jobs']);
  toast.success('Job added successfully ');
  return redirect('all-jobs');
} catch (error) {
  toast.error(error?.response?.data?.msg);
  return error;
}
};

```

```

const AddJob = () => {
  const { user } = useOutletContext();

  return (
    <Wrapper>
      <Form method='post' className='form'>
        <h4 className='form-title'>add job</h4>
        <div className='form-center'>
          <FormRow type='text' name='position' />
          <FormRow type='text' name='company' />
          <FormRow
            type='text'
            labelText='job location'
            name='jobLocation'

```

```
        defaultValue={user.location}
      />
      <FormRowSelect
        labelText='job status'
        name='jobStatus'
        defaultValue={JOB_STATUS.PENDING}
        list={Object.values(JOB_STATUS)}
      />
      <FormRowSelect
        labelText='job type'
        name='jobType'
        defaultValue={JOB_TYPE.FULL_TIME}
        list={Object.values(JOB_TYPE)}
      />
      <SubmitBtn formBtn />
    </div>
  </Form>
</Wrapper>
);
};
export default AddJob;
```

```
=====
File: client/src/pages/Admin.jsx
=====
```

```
import { FaSuitcaseRolling, FaCalendarCheck } from 'react-icons/fa';
import { useLoaderData, redirect } from 'react-router-dom';
import customFetch from '../utils/customFetch';
import Wrapper from '../assets/wrappers/StatsContainer';
import { toast } from 'react-toastify';
import { StatItem } from '../components';
```

```
export const loader = async () => {
  try {
    const response = await customFetch.get('/users/admin/app-stats');
    return response.data;
  } catch (error) {
    toast.error('You are not authorized to view this page');
    return redirect('/dashboard');
  }
};
```

```
const Admin = () => {
  const { users, jobs } = useLoaderData();
  return (
    <Wrapper>
      <StatItem
        title='current users'
        count={users}
        color='#e9b949'
        bcg='#fcefc7'
```

```

        icon={<FaSuitcaseRolling />}
      />
    <StatItem
      title='total jobs'
      count={jobs}
      color='#647acb'
      bcg='#e0e8f9'
      icon={<FaCalendarCheck />}
    />
  </Wrapper>
);
};
export default Admin;

```

```
=====
```

File: client/src/pages/AllJobs.jsx

```
=====
```

```

import { toast } from 'react-toastify';
import { JobsContainer, SearchContainer } from '../components';
import customFetch from '../utils/customFetch';
import { useLoaderData } from 'react-router-dom';
import { useContext, createContext } from 'react';
import { useQuery } from '@tanstack/react-query';

const allJobsQuery = (params) => {

```

```
const { search, jobStatus, jobType, sort, page } = params;

return {
  queryKey: [
    'jobs',
    search ?? '',
    jobStatus ?? 'all',
    jobType ?? 'all',
    sort ?? 'newest',
    page ?? 1,
  ],
  queryFn: async () => {
    const { data } = await customFetch.get('/jobs', {
      params,
    });
    return data;
  },
};
```

```
export const loader =
  (queryClient) =>
    async ({ request }) => {
      const params = Object.fromEntries([
        ...new URL(request.url).searchParams.entries(),
      ]);
```

```
    await queryClient.ensureQueryData(allJobsQuery(params));  
    return { searchValues: { ...params } };  
};
```

```
const AllJobsContext = createContext();  
  
const AllJobs = () => {  
  const { searchValues } = useLoaderData();  
  const { data } = useQuery(allJobsQuery(searchValues));  
  return (  
    <AllJobsContext.Provider value={{ data, searchValues }}>  
      <SearchContainer />  
      <JobsContainer />  
    </AllJobsContext.Provider>  
  );  
};
```

```
export const useAllJobsContext = () => useContext(AllJobsContext);
```

```
export default AllJobs;
```

```
=====
```

```
File: client/src/pages/DashboardLayout.jsx
```

```
=====
```

```
import { Outlet, redirect, useNavigate, useNavigation } from 'react-router-dom';  
import Wrapper from '../assets/wrappers/Dashboard';
```

```
import { BigSidebar, Navbar, SmallSidebar, Loading } from '../components';

import { createContext, useContext, useEffect, useState } from 'react';

import customFetch from '../utils/customFetch';

import { toast } from 'react-toastify';

import { useQuery } from '@tanstack/react-query';

import { checkDefaultTheme } from '../App';

const userQuery = {

  queryKey: ['user'],

  queryFn: async () => {

    const { data } = await customFetch.get('/users/current-user');

    return data;

  },

};

export const loader = (queryClient) => async () => {

  try {

    return await queryClient.ensureQueryData(userQuery);

  } catch (error) {

    return redirect('/');

  }

};

const DashboardContext = createContext();

const DashboardLayout = ({ queryClient }) => {

  const { user } = useQuery(userQuery).data;
```

```
const navigate = useNavigate();  
const navigation = useNavigation();  
const isPageLoading = navigation.state === 'loading';  
const [showSidebar, setShowSidebar] = useState(false);  
const [isDarkTheme, setIsDarkTheme] = useState(checkDefaultTheme());  
const [isAuthError, setIsAuthError] = useState(false);
```

```
const toggleDarkTheme = () => {  
  const newDarkTheme = !isDarkTheme;  
  setIsDarkTheme(newDarkTheme);  
  document.body.classList.toggle('dark-theme', newDarkTheme);  
  localStorage.setItem('darkTheme', newDarkTheme);  
};
```

```
const toggleSidebar = () => {  
  setShowSidebar(!showSidebar);  
};
```

```
const logoutUser = async () => {  
  navigate('/');  
  await customFetch.get('/auth/logout');  
  queryClient.invalidateQueries();  
  toast.success('Logging out...');  
};
```

```
customFetch.interceptors.response.use(
```



```
(response) => {  
  return response;  
},  
(error) => {  
  if (error?.response?.status === 401) {  
    setIsAuthError(true);  
  }  
  return Promise.reject(error);  
}  
);
```

```
useEffect(() => {  
  if (!isAuthError) return;  
  logoutUser();  
}, [isAuthError]);
```

```
return (  
  <DashboardContext.Provider  
    value={{  
      user,  
      showSidebar,  
      isDarkTheme,  
      toggleDarkTheme,  
      toggleSidebar,  
      logoutUser,  
    }}  
  )
```

```

>
<Wrapper>
  <main className='dashboard'>
    <SmallSidebar />
    <BigSidebar />
    <div>
      <Navbar />
      <div className='dashboard-page'>
        {isPageLoading ? <Loading /> : <Outlet context={{ user }} />}
      </div>
    </div>
  </main>
</Wrapper>
</DashboardContext.Provider>
);
};
export const useDashboardContext = () => useContext(DashboardContext);
export default DashboardLayout;

```

```
=====
```

File: client/src/pages/DeleteJob.jsx

```
=====
```

```

import { toast } from 'react-toastify';
import customFetch from '../utils/customFetch';
import { redirect } from 'react-router-dom';

```

```

export const action =
  (queryClient) =>
    async ({ params }) => {
      try {
        await customFetch.delete(`/jobs/${params.id}`);
        queryClient.invalidateQueries(['jobs']);

        toast.success('Job deleted successfully');
      } catch (error) {
        toast.error(error?.response?.data?.msg);
      }
      return redirect('/dashboard/all-jobs');
    };

```

=====

File: client/src/pages/EditJob.jsx

=====

```

import { FormRow, FormRowSelect, SubmitBtn } from '../components';
import Wrapper from '../assets/wrappers/DashboardFormPage';
import { useLoaderData, useParams } from 'react-router-dom';
import { JOB_STATUS, JOB_TYPE } from '../../utils/constants';
import { Form, redirect } from 'react-router-dom';
import { toast } from 'react-toastify';
import customFetch from '../utils/customFetch';

```

```
import { useQuery } from '@tanstack/react-query';
```

```
const singleJobQuery = (id) => {  
  return {  
    queryKey: ['job', id],  
    queryFn: async () => {  
      const { data } = await customFetch.get(`/jobs/${id}`);  
      return data;  
    },  
  };  
};
```

```
export const loader =  
  (queryClient) =>  
    async ({ params }) => {  
      try {  
        await queryClient.ensureQueryData(singleJobQuery(params.id));  
        return params.id;  
      } catch (error) {  
        toast.error(error?.response?.data?.msg);  
        return redirect('/dashboard/all-jobs');  
      }  
    };  
};
```

```
export const action =  
  (queryClient) =>  
    async ({ request, params }) => {
```

```
const formData = await request.formData();
const data = Object.fromEntries(formData);
try {
  await customFetch.patch(`/jobs/${params.id}`, data);
  queryClient.invalidateQueries(['jobs']);

  toast.success('Job edited successfully');
  return redirect('/dashboard/all-jobs');
} catch (error) {
  toast.error(error?.response?.data?.msg);
  return error;
}
};
```

```
const EditJob = () => {
  const id = useLoaderData();

  const {
    data: { job },
  } = useQuery(singleJobQuery(id));

  return (
    <Wrapper>
      <Form method='post' className='form'>
        <h4 className='form-title'>edit job</h4>
        <div className='form-center'>
```

```
<FormRow type='text' name='position' defaultValue={job.position} />
<FormRow type='text' name='company' defaultValue={job.company} />
<FormRow
  type='text'
  name='jobLocation'
  labelText='job location'
  defaultValue={job.jobLocation}
/>
<FormRowSelect
  name='jobStatus'
  labelText='job status'
  defaultValue={job.jobStatus}
  list={Object.values(JOB_STATUS)}
/>
<FormRowSelect
  name='jobType'
  labelText='job type'
  defaultValue={job.jobType}
  list={Object.values(JOB_TYPE)}
/>
<SubmitBtn formBtn />
</div>
</Form>
</Wrapper>
);
};
```

```
export default EditJob;
```

```
=====
```

```
File: client/src/pages/Error.jsx
```

```
=====
```

```
import { Link, useRouteError } from 'react-router-dom';
import Wrapper from '../assets/wrappers/ErrorPage';
import img from '../assets/images/not-found.svg';
const Error = () => {
  const error = useRouteError();
  console.log(error);
  if (error.status === 404) {
    return (
      <Wrapper>
        <div>
          <img src={img} alt='not found' />
          <h3>Ohh! page not found</h3>
          <p>we can't seem to find the page you are looking for</p>
          <Link to='/dashboard'>back home</Link>
        </div>
      </Wrapper>
    );
  }
  return (
    <Wrapper>
```

```
    <div>

      <h3>something went wrong</h3>

    </div>

  </Wrapper>

);

};

export default Error;
```

=====

File: client/src/pages/HomeLayout.jsx

=====

```
import { Outlet } from 'react-router-dom';

const HomeLayout = () => {

  return (

    <>

      <Outlet />

    </>

  );

};

export default HomeLayout;
```

=====

File: client/src/pages/Landing.jsx

=====


```
import styled from 'styled-components';
import Wrapper from '../assets/wrappers/LandingPage';
import main from '../assets/images/main.svg';
import { Link } from 'react-router-dom';
import { Logo } from '../components';

const Landing = () => {
  return (
    <Wrapper>
      <nav>
        <Logo />
      </nav>
      <div className='container page'>
        <div className='info'>
          <h1>
            job <span>tracking</span> app
          </h1>
          <p>
            I'm baby wayfarers hoodie next level taiyaki brooklyn cliché blue
            bottle single-origin coffee chia. Aesthetic post-ironic venmo,
            quinoa lo-fi tote bag adaptogen everyday carry meggings +1 brunch
            narwhal.
          </p>
          <Link to='/register' className='btn register-link'>
            Register
          </Link>
        </div>
      </div>
    </Wrapper>
  );
}
```

```

    <Link to='/login' className='btn '>
      Login / Demo User
    </Link>
  </div>

  <img src={main} alt='job hunt' className='img main-img' />
</div>

</Wrapper>

);
};

```

```
export default Landing;
```

```
=====
```

```
File: client/src/pages/Login.jsx
```

```
=====
```

```

import { Link, Form, redirect, useNavigate } from 'react-router-dom';
import Wrapper from '../assets/wrappers/RegisterAndLoginPage';
import { FormRow, Logo, SubmitBtn } from '../components';
import customFetch from '../utils/customFetch';
import { toast } from 'react-toastify';

```

```

export const action =
  (queryClient) =>
  async ({ request }) => {
    const formData = await request.formData();

```

```
const data = Object.fromEntries(formData);

try {

  await customFetch.post('/auth/login', data);

  queryClient.invalidateQueries();

  toast.success('Login successful');

  return redirect('/dashboard');

} catch (error) {

  toast.error(error?.response?.data?.msg);

  return error;

}

};
```

```
const Login = () => {
```

```
  const navigate = useNavigate();
```

```
  const loginDemoUser = async () => {
```

```
    const data = {
```

```
      email: 'test@test.com',
```

```
      password: 'secret123',
```

```
    };
```

```
    try {
```

```
      await customFetch.post('/auth/login', data);
```

```
      toast.success('Take a test drive');
```

```
      navigate('/dashboard');
```

```
    } catch (error) {
```

```
      toast.error(error?.response?.data?.msg);
```

```
    }  
  };  
  return (  
    <Wrapper>  
      <Form method='post' className='form'>  
        <Logo />  
        <h4>login</h4>  
        <FormRow type='email' name='email' />  
        <FormRow type='password' name='password' />  
        <SubmitBtn />  
        <button type='button' className='btn btn-block' onClick={loginDemoUser}>  
          explore the app  
        </button>  
        <p>  
          Not a member yet?  
          <Link to='/register' className='member-btn'>  
            Register  
          </Link>  
        </p>  
      </Form>  
    </Wrapper>  
  );  
};  
export default Login;
```

=====

File: client/src/pages/Profile.jsx

=====

```
import { FormRow, SubmitBtn } from '../components';
import Wrapper from '../assets/wrappers/DashboardFormPage';
import { useOutletContext, redirect } from 'react-router-dom';
import { Form } from 'react-router-dom';
import customFetch from '../utils/customFetch';
import { toast } from 'react-toastify';
```

```
export const action =
(queryClient) =>
async ({ request }) => {
  const formData = await request.formData();
  const file = formData.get('avatar');
  if (file && file.size > 500000) {
    toast.error('Image size too large');
    return null;
  }
  try {
    await customFetch.patch('/users/update-user', formData);
    queryClient.invalidateQueries(['user']);
    toast.success('Profile updated successfully');
    return redirect('/dashboard');
  } catch (error) {
    toast.error(error?.response?.data?.msg);
  }
}
```

```
    return null;
  }
};
```

```
const Profile = () => {
  const { user } = useOutletContext();

  const { name, lastName, email, location } = user;
```

```
  return (
    <Wrapper>
      <Form method='post' className='form' encType='multipart/form-data'>
        <h4 className='form-title'>profile</h4>
        <div className='form-center'>
          <div className='form-row'>
            <label htmlFor='avatar' className='form-label'>
              Select an image file (max 0.5 MB)
            </label>
            <input
              type='file'
              id='avatar'
              name='avatar'
              className='form-input'
              accept='image/*'
            />
          </div>
        </div>
```

```

    <FormRow type='text' name='name' defaultValue={name} />

    <FormRow
      type='text'
      name='lastName'
      labelText='last name'
      defaultValue={lastName}
    />

    <FormRow type='email' name='email' defaultValue={email} />
    <FormRow type='text' name='location' defaultValue={location} />
    <SubmitBtn formBtn />

  </div>

</Form>

</Wrapper>

);

};

export default Profile;

```

```

=====

File: client/src/pages/Register.jsx

=====

import { Form, redirect, Link } from 'react-router-dom';
import Wrapper from '../assets/wrappers/RegisterAndLoginPage';
import { FormRow, Logo, SubmitBtn } from '../components';
import customFetch from '../utils/customFetch';
import { toast } from 'react-toastify';

```

```
export const action = async ({ request }) => {  
  const formData = await request.formData();  
  const data = Object.fromEntries(formData);  
  
  try {  
    await customFetch.post('/auth/register', data);  
    toast.success('Registration successful');  
    return redirect('/login');  
  } catch (error) {  
    toast.error(error?.response?.data?.msg);  
  
    return error;  
  }  
};  
  
const Register = () => {  
  return (  
    <Wrapper>  
      <Form method='post' className='form'>  
        <Logo />  
        <h4>Register</h4>  
        <FormRow type='text' name='name' />  
        <FormRow type='text' name='lastName' labelText='last name' />  
        <FormRow type='text' name='location' />  
        <FormRow type='email' name='email' />  
        <FormRow type='password' name='password' />  
        <SubmitBtn />  
      </Form>  
    </Wrapper>  
  );  
};
```



```

    <p>
      Already a member?
      <Link to='/login' className='member-btn'>
        Login
      </Link>
    </p>
  </Form>
</Wrapper>
);
};
export default Register;

```

```
=====
```

File: client/src/pages/Stats.jsx

```
=====
```

```

import { ChartsContainer, StatsContainer } from '../components';
import customFetch from '../utils/customFetch';
import { useLoaderData } from 'react-router-dom';
import { useQuery } from '@tanstack/react-query';

```

```

const statsQuery = {
  queryKey: ['stats'],
  queryFn: async () => {
    const response = await customFetch.get('/jobs/stats');
    return response.data;
  }
};

```

```

    },
  };

export const loader = (queryClient) => async () => {
  const data = await queryClient.ensureQueryData(statsQuery);
  return null;
};

const Stats = () => {
  const { data } = useQuery(statsQuery);
  const { defaultStats, monthlyApplications } = data;

  return (
    <>
      <StatsContainer defaultStats={defaultStats} />
      {monthlyApplications?.length > 1 && (
        <ChartsContainer data={monthlyApplications} />
      )}
    </>
  );
};

export default Stats;

```

=====

File: client/src/pages/index.js

```
=====

export { default as DashboardLayout } from './DashboardLayout';
export { default as Landing } from './Landing';
export { default as HomeLayout } from './HomeLayout';
export { default as Register } from './Register';
export { default as Login } from './Login';
export { default as Error } from './Error';
export { default as Stats } from './Stats';
export { default as AllJobs } from './AllJobs';
export { default as AddJob } from './AddJob';
export { default as EditJob } from './EditJob';
export { default as Profile } from './Profile';
export { default as Admin } from './Admin';
```

```
=====

File: client/src/Utils/customFetch.js
```

```
=====

import axios from 'axios';

const customFetch = axios.create({
  baseURL: '/api/v1',
});

export default customFetch;
```

```
=====
File: client/src/utls/links.jsx
=====
```

```
import React from 'react';
```

```
import { IoBarChartSharp } from 'react-icons/io5';
```

```
import { MdQueryStats } from 'react-icons/md';
```

```
import { FaWpforms } from 'react-icons/fa';
```

```
import { ImProfile } from 'react-icons/im';
```

```
import { MdAdminPanelSettings } from 'react-icons/md';
```

```
const links = [
```

```
{
```

```
  text: 'add job',
```

```
  path: '.',
```

```
  icon: <FaWpforms />,
```

```
},
```

```
{
```

```
  text: 'all jobs',
```

```
  path: 'all-jobs',
```

```
  icon: <MdQueryStats />,
```

```
},
```

```
{
```

```
  text: 'stats',
```

```
  path: 'stats',
```

```
    icon: <IoBarChartSharp />,
  },
  {
    text: 'profile',
    path: 'profile',
    icon: <ImProfile />,
  },
  {
    text: 'admin',
    path: 'admin',
    icon: <MdAdminPanelSettings />,
  },
];
```

```
export default links;
```

```
=====
```

```
File: controllers/authController.js
```

```
=====
```

```
import { StatusCodes } from 'http-status-codes';
import User from '../models/UserModel.js';
import { comparePassword, hashPassword } from '../utils/passwordUtils.js';
import { UnauthenticatedError } from '../errors/customErrors.js';
import { createJWT } from '../utils/tokenUtils.js';
```

```
export const register = async (req, res) => {  
  const isFirstAccount = (await User.countDocuments()) === 0;  
  req.body.role = isFirstAccount ? 'admin' : 'user';  
  
  const hashedPassword = await hashPassword(req.body.password);  
  req.body.password = hashedPassword;  
  
  const user = await User.create(req.body);  
  res.status(StatusCodes.CREATED).json({ msg: 'user created' });  
};  
  
export const login = async (req, res) => {  
  const user = await User.findOne({ email: req.body.email });  
  
  const isValidUser =  
    user && (await comparePassword(req.body.password, user.password));  
  
  if (!isValidUser) throw new UnauthenticatedError('invalid credentials');  
  
  const token = createJWT({ userId: user._id, role: user.role });  
  
  const oneDay = 1000 * 60 * 60 * 24;  
  
  res.cookie('token', token, {  
    httpOnly: true,  
    expires: new Date(Date.now() + oneDay),  
    secure: process.env.NODE_ENV === 'production',
```

```

});

res.status(StatusCodes.OK).json({ msg: 'user logged in' });

};

export const logout = (req, res) => {

  res.cookie('token', 'logout', {

    httpOnly: true,

    expires: new Date(Date.now()),

  });

  res.status(StatusCodes.OK).json({ msg: 'user logged out!' });

};

```

```
=====
```

File: controllers/jobController.js

```
=====
```

```

import Job from '../models/JobModel.js';

import { StatusCodes } from 'http-status-codes';

import mongoose from 'mongoose';

import day from 'dayjs';

export const getAllJobs = async (req, res) => {

  const { search, jobStatus, jobType, sort } = req.query;

  const queryObject = {

    createdBy: req.user.userId,

```

```
};
```

```
if (search) {  
  queryObject.$or = [  
    { position: { $regex: search, $options: 'i' } },  
    { company: { $regex: search, $options: 'i' } },  
  ];  
}
```

```
if (jobStatus && jobStatus !== 'all') {  
  queryObject.jobStatus = jobStatus;  
}  
  
if (jobType && jobType !== 'all') {  
  queryObject.jobType = jobType;  
}
```

```
const sortOptions = {  
  newest: '-createdAt',  
  oldest: 'createdAt',  
  'a-z': 'position',  
  'z-a': '-position',  
};
```

```
const sortKey = sortOptions[sort] || sortOptions.newest;
```

```
// setup pagination
```



```
const page = Number(req.query.page) || 1;
const limit = Number(req.query.limit) || 10;
const skip = (page - 1) * limit;
```

```
const jobs = await Job.find(queryObject)
  .sort(sortKey)
  .skip(skip)
  .limit(limit);
```

```
const totalJobs = await Job.countDocuments(queryObject);
const numOfPages = Math.ceil(totalJobs / limit);
res
  .status(StatusCodes.OK)
  .json({ totalJobs, numOfPages, currentPage: page, jobs });
};
```

```
export const createJob = async (req, res) => {
  req.body.createdBy = req.user.userId;
  const job = await Job.create(req.body);
  res.status(StatusCodes.CREATED).json({ job });
};
```

```
export const getJob = async (req, res) => {
  const job = await Job.findById(req.params.id);
  res.status(StatusCodes.OK).json({ job });
};
```

```
};
```

```
export const updateJob = async (req, res) => {  
  const updatedJob = await Job.findByIdAndUpdate(req.params.id, req.body, {  
    new: true,  
  });
```

```
  res.status(StatusCodes.OK).json({ msg: 'job modified', job: updatedJob });  
};
```

```
export const deleteJob = async (req, res) => {  
  const removedJob = await Job.findByIdAndDelete(req.params.id);  
  res.status(StatusCodes.OK).json({ msg: 'job deleted', job: removedJob });  
};
```

```
export const showStats = async (req, res) => {  
  let stats = await Job.aggregate([  
    { $match: { createdBy: new mongoose.Types.ObjectId(req.user.userId) } },  
    { $group: { _id: '$jobStatus', count: { $sum: 1 } } },  
  ]);
```

```
  stats = stats.reduce((acc, curr) => {  
    const { _id: title, count } = curr;  
    acc[title] = count;  
    return acc;  
  }, {});
```

```
const defaultStats = {  
  pending: stats.pending || 0,  
  interview: stats.interview || 0,  
  declined: stats.declined || 0,  
};
```

```
let monthlyApplications = await Job.aggregate([  
  { $match: { createdBy: new mongoose.Types.ObjectId(req.user.userId) } },  
  {  
    $group: {  
      _id: { year: { $year: '$createdAt' }, month: { $month: '$createdAt' } },  
      count: { $sum: 1 },  
    },  
  },  
  { $sort: { '_id.year': -1, '_id.month': -1 } },  
  { $limit: 6 },  
]);
```

```
monthlyApplications = monthlyApplications  
  .map((item) => {  
    const {  
      _id: { year, month },  
      count,  
    } = item;
```

```

    const date = day()
      .month(month - 1)
      .year(year)
      .format('MMM YY');

    return { date, count };
  })
  .reverse();

res.status(StatusCodes.OK).json({ defaultStats, monthlyApplications });
};

```

```

=====

File: controllers/userController.js

=====

import { StatusCodes } from 'http-status-codes';
import User from '../models/UserModel.js';
import Job from '../models/JobModel.js';
import cloudinary from 'cloudinary';
import { formatImage } from '../middleware/multerMiddleware.js';

export const getCurrentUser = async (req, res) => {
  const user = await User.findOne({ _id: req.user.userId });
  const userWithoutPassword = user.toJSON();
  res.status(StatusCodes.OK).json({ user: userWithoutPassword });

```

```
};

export const getApplicationStats = async (req, res) => {

  const users = await User.countDocuments();

  const jobs = await Job.countDocuments();

  res.status(StatusCodes.OK).json({ users, jobs });

};

export const updateUser = async (req, res) => {

  const newUser = { ...req.body };

  delete newUser.password;

  delete newUser.role;


  if (req.file) {

    const file = formatImage(req.file);

    const response = await cloudinary.v2.uploader.upload(file);

    newUser.avatar = response.secure_url;

    newUser.avatarPublicId = response.public_id;

  }

  const updatedUser = await User.findByIdAndUpdate(req.user.userId, newUser);


  if (req.file && updatedUser.avatarPublicId) {

    await cloudinary.v2.uploader.destroy(updatedUser.avatarPublicId);

  }


  res.status(StatusCodes.OK).json({ msg: 'update user' });

};
```

```
=====
File: errors/customErrors.js
=====
```

```
import { StatusCodes } from 'http-status-codes';
```

```
export class NotFoundError extends Error {
```

```
  constructor(message) {
```

```
    super(message);
```

```
    this.name = 'NotFoundError';
```

```
    this.statusCode = StatusCodes.NOT_FOUND;
```

```
  }
```

```
}
```

```
export class BadRequestError extends Error {
```

```
  constructor(message) {
```

```
    super(message);
```

```
    this.name = 'BadRequestError';
```

```
    this.statusCode = StatusCodes.BAD_REQUEST;
```

```
  }
```

```
}
```

```
export class UnauthenticatedError extends Error {
```

```
  constructor(message) {
```

```
    super(message);
```

```
    this.name = 'UnauthenticatedError';
```

```
    this.statusCode = StatusCodes.UNAUTHORIZED;
```

```
  }
```

```
}  
  
export class UnauthorizedError extends Error {  
  constructor(message) {  
    super(message);  
    this.name = 'UnauthorizedError';  
    this.statusCode = StatusCodes.FORBIDDEN;  
  }  
}
```

=====

File: middleware/authMiddleware.js

=====

```
import {  
  UnauthenticatedError,  
  UnauthorizedError,  
  BadRequestError,  
} from '../errors/customErrors.js';  
  
import { verifyJWT } from '../utils/tokenUtils.js';  
  
export const authenticateUser = (req, res, next) => {  
  const { token } = req.cookies;  
  if (!token) throw new UnauthenticatedError('authentication invalid');  
  
  try {  
    const { userId, role } = verifyJWT(token);
```

```
const testUser = userId === '64b2c07ccac2efc972ab0eca';
req.user = { userId, role, testUser };
next();
} catch (error) {
  throw new UnauthenticatedError('authentication invalid');
}
};

export const authorizePermissions = (...roles) => {
  return (req, res, next) => {
    if (!roles.includes(req.user.role)) {
      throw new UnauthorizedError('Unauthorized to access this route');
    }
    next();
  };
};
```

```
export const checkForTestUser = (req, res, next) => {
  if (req.user.testUser) throw new BadRequestError('Demo User. Read Only!');
  next();
};
```

```
=====
File: middleware/errorHandlerMiddleware.js
=====
```



```
import { StatusCodes } from 'http-status-codes';
```

```
const errorHandlerMiddleware = (err, req, res, next) => {  
  console.log(err);  
  const statusCode = err.statusCode || StatusCodes.INTERNAL_SERVER_ERROR;  
  const msg = err.message || 'something went wrong, try again later';  
  res.status(statusCode).json({ msg });  
};
```

```
export default errorHandlerMiddleware;
```

```
=====
```

```
File: middleware/multerMiddleware.js
```

```
=====
```

```
import multer from 'multer';  
import DataParser from 'datauri/parser.js';  
import path from 'path';
```

```
const storage = multer.memoryStorage();
```

```
const upload = multer({ storage });
```

```
const parser = new DataParser();
```

```
export const formatImage = (file) => {
```

```
const fileExtension = path.extname(file.originalname).toString();  
return parser.format(fileExtension, file.buffer).content;  
};
```

```
export default upload;
```

```
=====
```

```
File: middleware/validationMiddleware.js
```

```
=====
```

```
import { body, param, validationResult } from 'express-validator';  
import {  
  BadRequestError,  
  NotFoundError,  
  UnauthorizedError,  
} from '../errors/customErrors.js';  
import { JOB_STATUS, JOB_TYPE } from '../utils/constants.js';  
import mongoose from 'mongoose';  
import Job from '../models/JobModel.js';  
import User from '../models/UserModel.js';
```

```
const withValidationErrors = (validateValues) => {  
  return [  
    validateValues,  
    (req, res, next) => {  
      const errors = validationResult(req);
```

```
if (!errors.isEmpty()) {  
  const errorMessages = errors.array().map((error) => error.msg);  
  
  const firstMessage = errorMessages[0];  
  console.log(Object.getPrototypeOf(firstMessage));  
  if (errorMessages[0].startsWith('no job')) {  
    throw new NotFoundError(errorMessages);  
  }  
  if (errorMessages[0].startsWith('not authorized')) {  
    throw new UnauthorizedError('not authorized to access this route');  
  }  
  throw new BadRequestError(errorMessages);  
}  
next();  
},  
];  
};
```

```
export const validateJobInput = withValidationErrors([  
  body('company').notEmpty().withMessage('company is required'),  
  body('position').notEmpty().withMessage('position is required'),  
  body('jobLocation').notEmpty().withMessage('job location is required'),  
  body('jobStatus')  
    .isIn(Object.values(JOB_STATUS))  
    .withMessage('invalid status value'),  
  body('jobType')
```

```
.isIn(Object.values(JOB_TYPE))
.withMessage('invalid type value'),
]);
```

```
export const validateIdParam = withValidationErrors([
  param('id').custom(async (value, { req }) => {
    const isValidMongold = mongoose.Types.ObjectId.isValid(value);
    if (!isValidMongold) throw new BadRequestError('invalid MongoDB id');
    const job = await Job.findById(value);
    if (!job) throw new NotFoundError(`no job with id ${value}`);
    const isAdmin = req.user.role === 'admin';
    const isOwner = req.user.userId === job.createdBy.toString();

    if (!isAdmin && !isOwner)
      throw new UnauthorizedError('not authorized to access this route');
  }),
]);
```

```
export const validateRegisterInput = withValidationErrors([
  body('name').notEmpty().withMessage('name is required'),
  body('email')
    .notEmpty()
    .withMessage('email is required')
    .isEmail()
    .withMessage('invalid email format')
    .custom(async (email) => {
```

```
const user = await User.findOne({ email });

if (user) {
  throw new BadRequestError('email already exists');
}

}),
body('password')
  .notEmpty()
  .withMessage('password is required')
  .isLength({ min: 8 })
  .withMessage('password must be at least 8 characters long'),
body('location').notEmpty().withMessage('location is required'),
body('lastName').notEmpty().withMessage('last name is required'),
]);
```

```
export const validateLoginInput = withValidationErrors([
  body('email')
    .notEmpty()
    .withMessage('email is required')
    .isEmail()
    .withMessage('invalid email format'),
  body('password').notEmpty().withMessage('password is required'),
]);
```

```
export const validateUpdateUserInput = withValidationErrors([
  body('name').notEmpty().withMessage('name is required'),
  body('email')
```

```

    .notEmpty()
    .withMessage('email is required')
    .isEmail()
    .withMessage('invalid email format')
    .custom(async (email, { req }) => {
      const user = await User.findOne({ email });
      if (user && user._id.toString() !== req.user.userId) {
        throw new BadRequestError('email already exists');
      }
    }),

    body('location').notEmpty().withMessage('location is required'),
    body('lastName').notEmpty().withMessage('last name is required'),
  ]);

```

```

=====
File: models/JobModel.js
=====

```

```

import mongoose from 'mongoose';
import { JOB_STATUS, JOB_TYPE } from '../utils/constants.js';
const JobSchema = new mongoose.Schema(
  {
    company: String,
    position: String,
    jobStatus: {

```

```
    type: String,
    enum: Object.values(JOB_STATUS),
    default: JOB_STATUS.PENDING,
  },
  jobType: {
    type: String,
    enum: Object.values(JOB_TYPE),
    default: JOB_TYPE.FULL_TIME,
  },
  jobLocation: {
    type: String,
    default: 'my city',
  },
  createdBy: {
    type: mongoose.Types.ObjectId,
    ref: 'User',
  },
},
{ timestamps: true }
);

export default mongoose.model('Job', JobSchema);
```

=====

File: models/UserModel.js

=====

```
import mongoose from 'mongoose';
```

```
const UserSchema = new mongoose.Schema({  
  name: String,  
  email: String,  
  password: String,  
  lastName: {  
    type: String,  
    default: 'lastName',  
  },  
  location: {  
    type: String,  
    default: 'my city',  
  },  
  role: {  
    type: String,  
    enum: ['user', 'admin'],  
    default: 'user',  
  },  
  avatar: String,  
  avatarPublicId: String,  
});
```

```
UserSchema.methods.toJSON = function () {  
  let obj = this.toObject();
```



```
delete obj.password;

return obj;

};
```

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

```
=====
```

File: routes/authRouter.js

```
=====
```

```
import { Router } from 'express';

const router = Router();

import { login, logout, register } from '../controllers/authController.js';

import {
  validateRegisterInput,
  validateLoginInput,
} from '../middleware/validationMiddleware.js';

import rateLimiter from 'express-rate-limit';

const apiLimiter = rateLimiter({
  windowMs: 15 * 60 * 1000, // 15 minutes
  max: 20,
  message: { msg: 'IP rate limit exceeded, retry in 15 minutes.' },
});
```

```
router.post('/register', apiLimiter, validateRegisterInput, register);
router.post('/login', apiLimiter, validateLoginInput, login);
router.get('/logout', logout);
```

```
export default router;
```

```
=====
```

File: routes/jobRouter.js

```
=====
```

```
import { Router } from 'express';
const router = Router();
import {
  getAllJobs,
  getJob,
  createJob,
  updateJob,
  deleteJob,
  showStats,
} from '../controllers/jobController.js';
import {
  validateJobInput,
  validateIdParam,
} from '../middleware/validationMiddleware.js';
import { checkForTestUser } from '../middleware/authMiddleware.js';
```

```
// router.get('/',getAllJobs)
```

```
// router.post('/',createJob)
```

```
router
```

```
.route('/')
```

```
.get(getAllJobs)
```

```
.post(checkForTestUser, validateJobInput, createJob);
```

```
router.route('/stats').get(showStats);
```

```
router
```

```
.route('/:id')
```

```
.get(validateIdParam, getJob)
```

```
.patch(checkForTestUser, validateJobInput, validateIdParam, updateJob)
```

```
.delete(checkForTestUser, validateIdParam, deleteJob);
```

```
export default router;
```

```
=====
```

```
File: routes/userRouter.js
```

```
=====
```

```
import { Router } from 'express';
```

```
import {
```

```
  getApplicationStats,
```

```
  getCurrentUser,
```

```
    updateUser,
  } from '../controllers/userController.js';
import { validateUpdateUserInput } from '../middleware/validationMiddleware.js';
import {
  authorizePermissions,
  checkForTestUser,
} from '../middleware/authMiddleware.js';
import upload from '../middleware/multerMiddleware.js';
const router = Router();

router.get('/current-user', getCurrentUser);
router.get('/admin/app-stats', [
  authorizePermissions('admin'),
  getApplicationStats,
]);
router.patch(
  '/update-user',
  checkForTestUser,
  upload.single('avatar'),
  validateUpdateUserInput,
  updateUser
);

export default router;
```

=====

File: utils/constants.js

=====

```
export const JOB_STATUS = {  
  PENDING: 'pending',  
  INTERVIEW: 'interview',  
  DECLINED: 'declined',  
};
```

```
export const JOB_TYPE = {  
  FULL_TIME: 'full-time',  
  PART_TIME: 'part-time',  
  INTERNSHIP: 'internship',  
};
```

```
export const JOB_SORT_BY = {  
  NEWEST_FIRST: 'newest',  
  OLDEST_FIRST: 'oldest',  
  ASCENDING: 'a-z',  
  DESCENDING: 'z-a',  
};
```

=====

File: utils/mockData.json

=====

{ "company": "Skinte", "position": "Senior Financial Analyst", "jobLocation": "Tonshayevo", "jobStatus": "pending", "jobType": "internship", "createdAt": "2022-09-25T00:02:40Z" },

{ "company": "Livefish", "position": "VP Marketing", "jobLocation": "Vale de Açores", "jobStatus": "declined", "jobType": "internship", "createdAt": "2023-04-16T22:34:17Z" },

{ "company": "Demimbu", "position": "Administrative Officer", "jobLocation": "Palmar de Varela", "jobStatus": "interview", "jobType": "internship", "createdAt": "2022-10-15T07:44:21Z" },

{ "company": "Gigashots", "position": "Account Representative III", "jobLocation": "Jazovo", "jobStatus": "pending", "jobType": "part-time", "createdAt": "2023-04-25T18:44:34Z" },

{ "company": "Feedspan", "position": "Automation Specialist IV", "jobLocation": "Botoh", "jobStatus": "interview", "jobType": "full-time", "createdAt": "2023-07-07T09:29:32Z" },

{ "company": "Camimbo", "position": "Accountant III", "jobLocation": "Tsapêrai", "jobStatus": "declined", "jobType": "full-time", "createdAt": "2023-04-04T08:46:19Z" },

{ "company": "Avamm", "position": "Financial Analyst", "jobLocation": "Kobe", "jobStatus": "interview", "jobType": "part-time", "createdAt": "2022-09-19T06:17:02Z" },

{ "company": "Dynazzy", "position": "Junior Executive", "jobLocation": "Uliastay", "jobStatus": "declined", "jobType": "full-time", "createdAt": "2023-03-17T16:19:19Z" },

{ "company": "Youspan", "position": "Assistant Media Planner", "jobLocation": "Mumias", "jobStatus": "interview", "jobType": "full-time", "createdAt": "2023-01-16T05:28:12Z" },

{ "company": "Shuffletag", "position": "VP Sales", "jobLocation": "Quangang", "jobStatus": "interview", "jobType": "internship", "createdAt": "2022-08-31T08:58:00Z" },

{ "company": "Npath", "position": "Desktop Support Technician", "jobLocation": "Arzakan", "jobStatus": "interview", "jobType": "internship", "createdAt": "2023-06-06T10:08:31Z" },

{"company":"Meevee","position":"Analyst Programmer","jobLocation":"Pršovce","jobStatus":"declined","jobType":"full-time","createdAt":"2023-06-05T05:51:02Z"},

{"company":"Meembee","position":"Office Assistant IV","jobLocation":"Radashkovichy","jobStatus":"pending","jobType":"full-time","createdAt":"2023-04-11T20:20:22Z"},

{"company":"Twitternation","position":"Speech Pathologist","jobLocation":"Agen","jobStatus":"declined","jobType":"part-time","createdAt":"2023-02-18T07:00:55Z"},

{"company":"Meejo","position":"Pharmacist","jobLocation":"Deshan","jobStatus":"pending","jobType":"internship","createdAt":"2023-02-02T04:00:43Z"},

{"company":"Wikizz","position":"Safety Technician IV","jobLocation":"Ube","jobStatus":"pending","jobType":"part-time","createdAt":"2022-11-05T09:32:23Z"},

{"company":"Muxo","position":"Senior Developer","jobLocation":"São Sepé","jobStatus":"interview","jobType":"internship","createdAt":"2023-05-03T04:09:56Z"},

{"company":"Meevee","position":"Actuary","jobLocation":"Manturovo","jobStatus":"interview","jobType":"part-time","createdAt":"2023-05-24T07:55:12Z"},

{"company":"Eadel","position":"Quality Control Specialist","jobLocation":"San Antonio","jobStatus":"interview","jobType":"internship","createdAt":"2022-12-03T20:21:49Z"},

{"company":"Mydo","position":"Physical Therapy Assistant","jobLocation":"Madīnat ash Shamāl","jobStatus":"pending","jobType":"part-time","createdAt":"2023-02-28T23:10:51Z"},

{"company":"Oba","position":"Social Worker","jobLocation":"Quinta dos Frades","jobStatus":"interview","jobType":"internship","createdAt":"2022-11-24T15:12:17Z"},

{"company":"Photobug","position":"Payment Adjustment Coordinator","jobLocation":"Chengdong","jobStatus":"pending","jobType":"full-time","createdAt":"2023-04-05T23:51:36Z"},

{"company":"Quimm","position":"Structural Engineer","jobLocation":"Strasbourg","jobStatus":"pending","jobType":"part-time","createdAt":"2022-09-26T20:00:08Z"},

```
{"company":"Flashdog","position":"Senior Editor","jobLocation":"Wilmington","jobStatus":"declined","jobType":"full-time","createdAt":"2023-02-19T17:59:12Z"},

{"company":"Fatz","position":"Help Desk Technician","jobLocation":"Verona","jobStatus":"pending","jobType":"part-time","createdAt":"2022-07-12T13:10:32Z"},

{"company":"Youopia","position":"Staff Accountant IV","jobLocation":"Río Tercero","jobStatus":"declined","jobType":"full-time","createdAt":"2022-07-12T00:04:36Z"},

{"company":"Abata","position":"Analyst Programmer","jobLocation":"Araguari","jobStatus":"declined","jobType":"full-time","createdAt":"2022-09-28T19:05:28Z"},

{"company":"Ntag","position":"Compensation Analyst","jobLocation":"Kowingir","jobStatus":"pending","jobType":"part-time","createdAt":"2023-06-18T08:52:02Z"},

{"company":"Bluejam","position":"Statistician I","jobLocation":"Levallois-Perret","jobStatus":"interview","jobType":"internship","createdAt":"2022-09-28T04:02:05Z"},

{"company":"Skynoodle","position":"Programmer II","jobLocation":"Dallas","jobStatus":"pending","jobType":"part-time","createdAt":"2022-08-03T19:41:57Z"},

{"company":"Zoovu","position":"Occupational Therapist","jobLocation":"Jiaocun","jobStatus":"declined","jobType":"full-time","createdAt":"2023-05-12T08:05:46Z"},

{"company":"Wordware","position":"Associate Professor","jobLocation":"Tanumshede","jobStatus":"declined","jobType":"part-time","createdAt":"2023-01-13T04:42:33Z"},

{"company":"Skippad","position":"Project Manager","jobLocation":"Pantai","jobStatus":"interview","jobType":"internship","createdAt":"2022-07-15T16:38:48Z"},

{"company":"Devbug","position":"Recruiter","jobLocation":"Basen","jobStatus":"declined","jobType":"part-time","createdAt":"2022-07-17T22:42:40Z"},

{"company":"Eabox","position":"Programmer II","jobLocation":"An Lão","jobStatus":"interview","jobType":"part-time","createdAt":"2023-01-08T16:27:44Z"},
```



```
{"company":"Thoughtstorm","position":"Occupational  
Therapist","jobLocation":"Tari","jobStatus":"declined","jobType":"part-  
time","createdAt":"2023-01-23T03:57:18Z"},  
  
{"company":"Wikido","position":"Recruiting  
Manager","jobLocation":"Pontian","jobStatus":"interview","jobType":"internship","createdAt":  
"2022-12-12T08:53:15Z"},  
  
{"company":"Midel","position":"Financial  
Advisor","jobLocation":"Finspång","jobStatus":"declined","jobType":"part-  
time","createdAt":"2022-12-19T14:02:56Z"},  
  
{"company":"Dynabox","position":"Human Resources  
Manager","jobLocation":"Pashiya","jobStatus":"pending","jobType":"internship","createdAt":"  
2022-09-09T22:32:52Z"},  
  
{"company":"Brainbox","position":"Software Test Engineer III","jobLocation":"Krajan  
Siki","jobStatus":"pending","jobType":"full-time","createdAt":"2022-08-22T10:10:58Z"},  
  
{"company":"Devbug","position":"Web Designer  
I","jobLocation":"Baisha","jobStatus":"interview","jobType":"full-time","createdAt":"2022-08-  
22T10:45:19Z"},  
  
{"company":"Trupe","position":"Geological Engineer","jobLocation":"El  
Cubolero","jobStatus":"declined","jobType":"internship","createdAt":"2022-12-  
27T18:49:49Z"},  
  
{"company":"Quire","position":"Safety Technician  
IV","jobLocation":"Anastácio","jobStatus":"pending","jobType":"part-time","createdAt":"2022-  
12-23T12:33:46Z"},  
  
{"company":"Jaxnation","position":"Staff Accountant  
IV","jobLocation":"Delanggu","jobStatus":"pending","jobType":"internship","createdAt":"2023  
-04-24T17:11:13Z"},  
  
{"company":"Skyble","position":"Internal  
Auditor","jobLocation":"Pristina","jobStatus":"declined","jobType":"full-  
time","createdAt":"2023-02-01T05:15:56Z"},  
  
{"company":"Gevee","position":"Sales  
Representative","jobLocation":"Huineneno","jobStatus":"declined","jobType":"internship","crea  
tedAt":"2022-10-30T06:45:32Z"},
```

{"company":"Thoughtmix","position":"Media Manager IV","jobLocation":"Unity","jobStatus":"interview","jobType":"full-time","createdAt":"2022-12-09T15:03:18Z"},

{"company":"Fivespan","position":"Mechanical Systems Engineer","jobLocation":"Sua","jobStatus":"interview","jobType":"internship","createdAt":"2022-09-05T20:44:56Z"},

{"company":"LiveZ","position":"Computer Systems Analyst II","jobLocation":"Huanggong","jobStatus":"declined","jobType":"full-time","createdAt":"2022-09-06T17:06:56Z"},

{"company":"Topicware","position":"Statistician I","jobLocation":"Ngilengan","jobStatus":"pending","jobType":"full-time","createdAt":"2023-04-29T22:25:49Z"},

{"company":"Ntag","position":"VP Quality Control","jobLocation":"Florescia","jobStatus":"declined","jobType":"internship","createdAt":"2022-07-11T05:54:35Z"},

{"company":"Digitube","position":"Senior Sales Associate","jobLocation":"Mibu","jobStatus":"declined","jobType":"internship","createdAt":"2022-09-06T16:24:05Z"},

{"company":"Thoughtbeat","position":"Associate Professor","jobLocation":"Bęczarka","jobStatus":"interview","jobType":"internship","createdAt":"2023-06-08T04:19:45Z"},

{"company":"Mynte","position":"GIS Technical Architect","jobLocation":"Palompon","jobStatus":"interview","jobType":"part-time","createdAt":"2022-11-21T04:11:59Z"},

{"company":"Midel","position":"Business Systems Development Analyst","jobLocation":"Waso","jobStatus":"declined","jobType":"full-time","createdAt":"2023-04-25T23:15:23Z"},

{"company":"Twiyoo","position":"Professor","jobLocation":"Cigalontang","jobStatus":"pending","jobType":"full-time","createdAt":"2022-11-29T17:03:56Z"},

{"company":"Meembee","position":"Administrative Officer","jobLocation":"Bavorov","jobStatus":"interview","jobType":"full-time","createdAt":"2023-06-17T05:21:49Z"},

{"company":"Avavee","position":"Safety Technician II","jobLocation":"Merdeka","jobStatus":"declined","jobType":"internship","createdAt":"2022-08-09T23:39:20Z"},

{"company":"Camido","position":"Associate Professor","jobLocation":"Grytviken","jobStatus":"interview","jobType":"full-time","createdAt":"2023-01-05T12:50:09Z"},

{"company":"Skyba","position":"Professor","jobLocation":"Sibubuhan","jobStatus":"declined","jobType":"full-time","createdAt":"2023-03-25T10:56:49Z"},

{"company":"Thoughtmix","position":"Administrative Assistant IV","jobLocation":"Gandu","jobStatus":"declined","jobType":"full-time","createdAt":"2023-04-04T16:19:08Z"},

{"company":"Thoughtbeat","position":"Actuary","jobLocation":"Paris 09","jobStatus":"pending","jobType":"full-time","createdAt":"2022-12-11T09:24:39Z"},

{"company":"Twitterlist","position":"Business Systems Development Analyst","jobLocation":"Dagou","jobStatus":"pending","jobType":"part-time","createdAt":"2022-10-21T10:08:02Z"},

{"company":"Divape","position":"Actuary","jobLocation":"Des Moines","jobStatus":"pending","jobType":"internship","createdAt":"2023-05-17T05:53:45Z"},

{"company":"Kayveo","position":"Food Chemist","jobLocation":"Tene","jobStatus":"interview","jobType":"part-time","createdAt":"2022-08-07T17:22:15Z"},

{"company":"Jabbersphere","position":"Occupational Therapist","jobLocation":"Jiayuguan","jobStatus":"pending","jobType":"full-time","createdAt":"2023-01-27T07:51:21Z"},

{"company":"Demimbu","position":"Media Manager I","jobLocation":"Navotas","jobStatus":"interview","jobType":"full-time","createdAt":"2022-11-01T08:43:56Z"},

{"company":"Cogilith","position":"Recruiting Manager","jobLocation":"Kemiri","jobStatus":"pending","jobType":"part-time","createdAt":"2023-05-17T21:01:01Z"},

{"company":"Dabtype","position":"Pharmacist","jobLocation":"Barranquilla","jobStatus":"declined","jobType":"internship","createdAt":"2022-11-25T10:11:06Z"},

{"company":"Twitterwire","position":"Programmer Analyst II","jobLocation":"Jangkungkusumo","jobStatus":"declined","jobType":"internship","createdAt":"2022-11-09T19:12:11Z"},

{"company":"Oba","position":"Human Resources Assistant III","jobLocation":"Fontainhas","jobStatus":"declined","jobType":"internship","createdAt":"2023-04-04T00:06:28Z"},

{"company":"Oloo","position":"Senior Financial Analyst","jobLocation":"Cabeças Verdes","jobStatus":"declined","jobType":"internship","createdAt":"2023-05-22T08:15:39Z"},

{"company":"Edgewire","position":"Electrical Engineer","jobLocation":"Jianping","jobStatus":"declined","jobType":"full-time","createdAt":"2022-12-18T09:50:49Z"},

{"company":"Zazio","position":"Chemical Engineer","jobLocation":"Linköping","jobStatus":"interview","jobType":"internship","createdAt":"2023-07-04T09:45:27Z"},

{"company":"Linkbuzz","position":"Research Nurse","jobLocation":"Nytva","jobStatus":"declined","jobType":"internship","createdAt":"2022-12-28T03:53:15Z"},

{"company":"Skyndu","position":"Senior Sales Associate","jobLocation":"Kendung Timur","jobStatus":"declined","jobType":"full-time","createdAt":"2023-06-22T21:37:51Z"},

{"company":"Livetube","position":"General Manager","jobLocation":"Tegarenkrajan","jobStatus":"declined","jobType":"part-time","createdAt":"2023-02-26T14:53:31Z"},

{"company":"Babbleopia","position":"Health Coach I","jobLocation":"Huancheng","jobStatus":"interview","jobType":"full-time","createdAt":"2023-05-02T06:19:25Z"},

{"company":"Dabjam","position":"Dental Hygienist","jobLocation":"Trpinja","jobStatus":"declined","jobType":"internship","createdAt":"2023-02-17T01:35:08Z"},

{"company":"Tazz","position":"Project Manager","jobLocation":"Caacupé","jobStatus":"declined","jobType":"full-time","createdAt":"2023-03-20T22:34:44Z"},

{"company":"Meejo","position":"Registered Nurse","jobLocation":"Santa Fé do Sul","jobStatus":"declined","jobType":"full-time","createdAt":"2023-05-02T03:44:18Z"},

{"company":"Omba","position":"Geologist III","jobLocation":"Biliri","jobStatus":"pending","jobType":"part-time","createdAt":"2022-09-29T04:05:34Z"},

{"company":"Thoughtstorm","position":"Design Engineer","jobLocation":"Damu","jobStatus":"pending","jobType":"full-time","createdAt":"2022-11-08T17:01:10Z"},

{"company":"Voolith","position":"General Manager","jobLocation":"Kuala Terengganu","jobStatus":"declined","jobType":"full-time","createdAt":"2023-01-31T11:40:44Z"},

{"company":"Wordpedia","position":"Software Consultant","jobLocation":"Ashibetsu","jobStatus":"declined","jobType":"part-time","createdAt":"2022-08-03T14:48:48Z"},

{"company":"Gevee","position":"Clinical Specialist","jobLocation":"Wujiashan","jobStatus":"interview","jobType":"part-time","createdAt":"2023-07-08T07:21:54Z"},

{"company":"Twitterlist","position":"Operator","jobLocation":"Mikhaylovsk","jobStatus":"pending","jobType":"part-time","createdAt":"2023-03-22T17:42:55Z"},

{"company":"Tekfly","position":"Research Assistant I","jobLocation":"Patarrá","jobStatus":"pending","jobType":"internship","createdAt":"2023-03-23T14:17:24Z"},

{"company":"Plambee","position":"Analog Circuit Design manager","jobLocation":"Les Coteaux","jobStatus":"declined","jobType":"internship","createdAt":"2022-11-24T07:53:51Z"},

{"company":"Roombo","position":"Community Outreach Specialist","jobLocation":"Yanglinshi","jobStatus":"interview","jobType":"part-time","createdAt":"2023-05-30T07:03:17Z"},

{"company":"Abata","position":"Account Representative II","jobLocation":"Shixi","jobStatus":"declined","jobType":"part-time","createdAt":"2023-03-27T22:14:04Z"},

{"company":"Layo","position":"Junior Executive","jobLocation":"Ikongo","jobStatus":"interview","jobType":"full-time","createdAt":"2022-09-24T20:05:33Z"},

```
{ "company": "DabZ", "position": "Staff Accountant",  
  "jobLocation": "Mandiana", "jobStatus": "declined", "jobType": "full-time", "createdAt": "2023-04-14T09:42:11Z"},  
  
{ "company": "Devify", "position": "Operator", "jobLocation": "Tabalosos", "jobStatus": "declined",  
  "jobType": "part-time", "createdAt": "2022-10-25T14:23:45Z"},  
  
{ "company": "Blognation", "position": "Social Worker", "jobLocation": "Křenovice", "jobStatus": "declined", "jobType": "full-time",  
  "createdAt": "2023-06-06T18:39:47Z"},  
  
{ "company": "Topdrive", "position": "Graphic Designer", "jobLocation": "Xinquansi", "jobStatus": "pending", "jobType": "full-time",  
  "createdAt": "2022-12-30T23:14:12Z"},  
  
{ "company": "Zooxo", "position": "Human Resources Assistant III", "jobLocation": "Gaotuo", "jobStatus": "interview", "jobType": "full-time", "createdAt": "2022-11-19T04:23:28Z"},  
  
{ "company": "Demimbu", "position": "Data Coordinator", "jobLocation": "Al Buṭayḥah", "jobStatus": "interview", "jobType": "internship", "createdAt": "2023-06-12T12:29:54Z"},  
  
{ "company": "Zava", "position": "Graphic Designer", "jobLocation": "Calebasses", "jobStatus": "interview", "jobType": "internship", "createdAt": "2022-08-03T14:02:24Z"},  
  
{ "company": "Janyx", "position": "Budget/Accounting Analyst I", "jobLocation": "Kosino", "jobStatus": "interview", "jobType": "part-time", "createdAt": "2022-09-30T01:02:16Z"}]
```

=====

File: utils/passwordUtils.js

=====

```
import bcrypt from 'bcryptjs';  
  
export const hashPassword = async (password) => {  
  const salt = await bcrypt.genSalt(10);
```

```
const hashedPassword = await bcrypt.hash(password, salt);

return hashedPassword;

};

export const comparePassword = async (password, hashedPassword) => {

const isMatch = await bcrypt.compare(password, hashedPassword);

return isMatch;

};
```

=====

File: utils/tokenUtils.js

=====

```
import jwt from 'jsonwebtoken';

export const createJWT = (payload) => {

const token = jwt.sign(payload, process.env.JWT_SECRET, {

expiresIn: process.env.JWT_EXPIRES_IN,

});

return token;

};

export const verifyJWT = (token) => {

const decoded = jwt.verify(token, process.env.JWT_SECRET);

return decoded;

};
```

