

 **Project Title:**

**DevConnect** – A Full-Stack Developer Networking Platform

---

 **Abstract:**

**DevConnect** is a full-stack web application that allows developers to register, log in, create profiles, and connect with others. It's a mini-developer community platform focused on core authentication, protected routes, REST API usage, and modern frontend development using React and Redux.

This platform demonstrates seamless integration between the frontend and backend, state management, and secure data handling. Built using the **MERN stack** (MongoDB, Express, React, Node.js), it serves as a foundational model for building secure and scalable web applications.

---

 **Objectives:**

- Build a secure user authentication system.
  - Create an intuitive and responsive frontend using React.
  - Implement a RESTful backend with protected API routes.
  - Design a user-friendly dashboard and profile system.
  - Practice integration of Redux for global state management.
  - Demonstrate knowledge of full-stack development life cycle.
- 

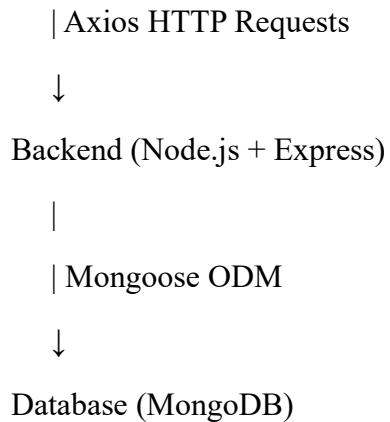
 **Architecture Overview:**

yaml

CopyEdit

Frontend (React + Redux)

|



### 🔧 Technologies Used:

#### Category Tech Stack

**Frontend** React, Redux, React Router DOM

**Backend** Node.js, Express.js

**Database** MongoDB with Mongoose

**Auth** JSON Web Tokens (JWT), bcryptjs

**Versioning** Git & GitHub

**Dev Tools** Postman, VS Code, npm

---

### 📁 Folder Structure Summary:

plaintext

CopyEdit

devconnect/

  └── frontend/ (React App)

  └── backend/ (Express API)

  └── README.md

  └── package.json

---

## Key Features:

-  User registration & login with password hashing
  -  Protected routes using JWT
  -  Profile data fetched from MongoDB
  -  Personalized dashboard after login
  -  State management with Redux
  -  Clean UI with navigation using React Router
- 

## Authentication Flow:

1. **Register:** User submits name, email, password.
    - Backend hashes the password using bcrypt and stores the user in MongoDB.
  2. **Login:** User submits email & password.
    - Backend verifies credentials and returns a signed JWT token.
  3. **Protected Routes:** Token is stored in localStorage.
    - On each protected API request, token is sent via headers and validated.
- 

## How It Works (Flow Explanation):

1.  **Home Page:** Landing page with nav options
2.  **Register/Login:** Authentication with JWT
3.  **Frontend:**
  - React captures form data
  - Axios sends data to backend
  - Redux stores auth state
4.  **Backend:**
  - Express routes receive requests
  - Controllers validate input and respond with JWT

- Middleware checks for valid tokens

## 5. Database:

- MongoDB stores user data and profiles
  - Mongoose handles schema and operations
- 

## API Endpoints Summary:

### **POST /api/auth/register**

Registers a new user

### **POST /api/auth/login**

Authenticates user and returns JWT

### **GET /api/user/me**

Returns authenticated user data

### **GET /api/user/:id**

Fetches public profile by ID

---

## Learnings:

- Connecting frontend and backend securely
  - Handling async API calls and state
  - Managing secure tokens and sessions
  - Structuring large-scale projects
  - Writing modular, maintainable code
-

---

## 💻 Installation & Execution Guide:

### 1. Clone and Setup

bash

CopyEdit

```
git clone https://github.com/ketansonwane7781/devconnect.git
```

```
cd devconnect
```

### 2. Backend Setup

bash

CopyEdit

```
cd backend
```

```
npm install
```

```
touch .env
```

.env file content:

env

CopyEdit

PORT=5000

MONGO\_URI=mongodb://localhost:27017/devconnect

JWT\_SECRET=your\_secret\_key

Start backend:

bash

CopyEdit

```
npm start
```

### 3. Frontend Setup

bash

CopyEdit

cd ../frontend

npm install

npm start

---

#### Tips & Notes:

- MongoDB must be running locally or via MongoDB Atlas.
  - Ensure React and Node servers are running in separate terminals.
  - Use [Postman](#) to test API endpoints.
  - For production deployment, consider using platforms like **Render**, **Vercel**, or **Netlify**.
- 

#### Author

**Ketan Kailas Sonawane**

GitHub: [@ketansonwane7781](#)

---