

DevScript Blog App - Architecture and Flow

1. Architecture

The application is divided into two major parts:

A. Frontend (React.js)

- Built with Vite for faster development.
- Routing is handled using react-router-dom .
- Authentication: On login/signup, the JWT token is saved in localStorage .
- Role-based UI: The token is decoded to check if the user is admin or user to display different navigation options.

Main Pages:

- / → Home Page showing list of blogs.
- /blogs/:id → Detailed blog page.
- /login → Login page.
- /signup → Sign-up page.
- /admin → Admin dashboard (only for admins).

B. Backend (Node.js + Express.js)

- Authentication APIs: /api/auth/login , /api/auth/signup
- Blog APIs: /api/blogs , /api/blogs/:id
- JWT: Issued on successful login and sent in response.
- MongoDB: Stores user and blog data.

Middlewares:

- Token is verified on the server for protected routes.

2. Flow of the Application

1. User opens the app and lands on the Home page.
2. User can:
 - Browse blogs without logging in.
 - Click Login or Sign Up to authenticate.
3. On login/signup:
 - Token is generated and stored in localStorage .
 - Token contains id and role .
4. Navbar conditionally displays:
 - Login/Signup options if not logged in.
 - Dashboard + Logout if logged in.
 - Dashboard is shown only if role is admin .
5. Blogs are fetched via /api/blogs endpoint.
6. Clicking on a blog navigates to /blogs/:id page showing full content.
7. Admin users can access /admin dashboard.

-
-