

## **Social-media-backend**

### **Used tools and libraries:**

- (i) Docker
- (ii) Postgresql (ORDBMS)
- (iii) GraphQL prisma

### **SQL tables creation:**

- (i) Already defined in schema.prisma file (within the project).
- (ii) Follow configure steps below to setup model schema.

### **API Endpoints Documentation: (used graphql)**

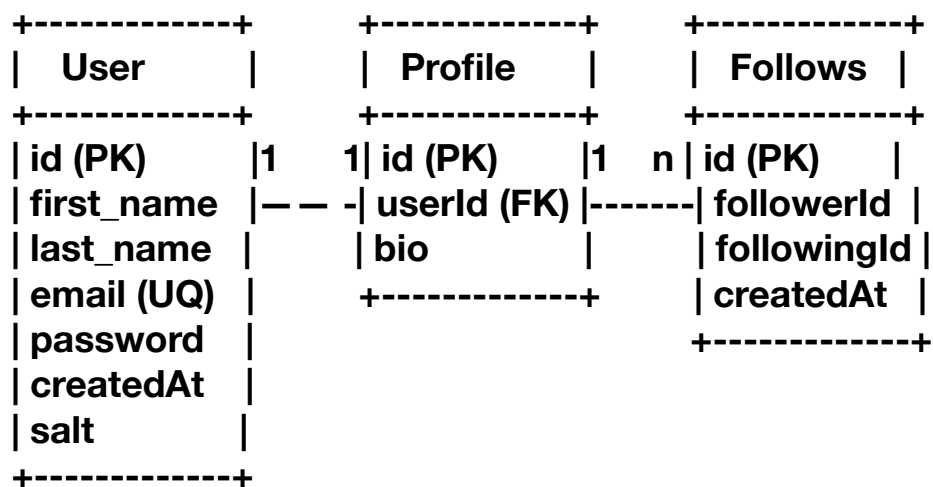
- (i) Since project is using graphql in its implementation , so upon opening, all the APIs will be available directly in the graphql UI(queries and mutations).
- (ii) Environment variable has been set up, if not provided, it will directly run on PORT 8001 by default.
- (iii) localhost://8001/graphql (or provide env).
- (iv) Authentication and Authorisation has been implemented.
- (v) Just run the project using command: node index.js and run it to check queries and mutations.
- (vi) Configure steps below to setup postgresql and create tables within docker container.

### **Configure Steps: (Used docker container for postgres)**

- (i) Open project. Add node\_modules
- (ii) Open your docker daemon terminal (or Docker Desktop)
- (iii) Run: docker compose up -d
- (iv) Open terminal -> run : docker exec - it <container\_id> bash
- (v) Run: su postgres
- (vi) Run: psql
- (vii) Run : \c socialmedia. —(now you are connected to db in container)
- (viii) Now go back to the project and run: npx prisma migrate dev - -name <tables\_created>
- (ix) Now you check the tables will be created using command : \d
- (x) Now run the project using command: node index.js
- (xi) Now you may implement graphql UI.

(xii) Begin with signUp.

### ER diagrams(Textual representation):



1  
|  
|  
n

