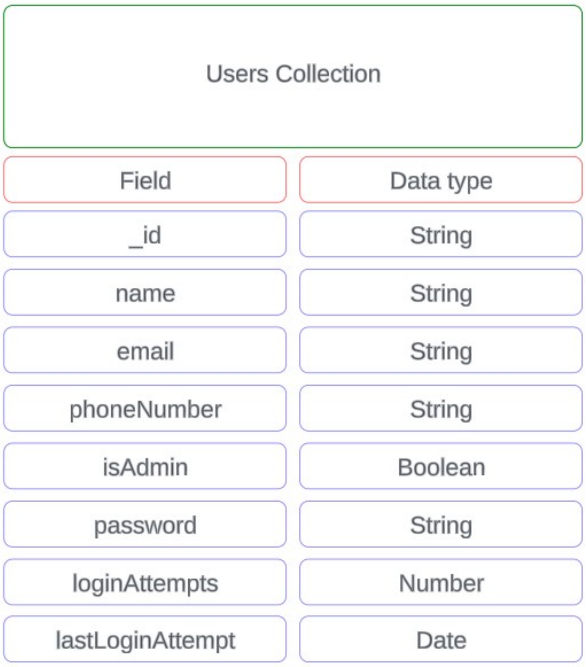
Authorization Service - Development Documentation

Database Design:



Users Collection

Field	Data type
_id	String
name	String
email	String
phoneNumber	String
isAdmin	Boolean
password	String
loginAttempts	Number
lastLoginAttempt	Date

API Endpoints:

- /api/login
- /api/signup

Features for the APIs:

- **Sign up:**
  - Check for any xss in input.
  - Validate the email string
  - Validate the phone number
  - Check for duplicate accounts
  - Check if strong password

- **Login:**
  - Check for xss in input
  - Check if the account exists
  - Check for correct password

- Send JWT token on successful login
- Limit invalid login attempts

Tech Stack:

Server: Node js , express js
Database: MongoDB
Packages used:

- **jsonwebtoken -** Used to make create and send jwt tokens can be used for session management
- **bcrypt** - Used to hash the passwords and compare the hashed passwords
- **xss -** Used to filter out xss on inputs
- **Mongoose -** ORM for monogoDB
- **validator -** Used to validate email and strong password
- **awsome-phonenumber -** Used to validate phone numbers.
- **serverless-http -** used to host the express server on serverless framework

Deployment:

- Installed global npm package serverless
- Setup a IAM user with required permissions for the deployment
- Configured serverless using accesskey from the IAM user using serverless config
- Configured the handler.js and serverless.yaml to prepare for deployment
- Addded the deploy command in package.json
- Deployed the APIs on AWS lambda using npm run deploy command