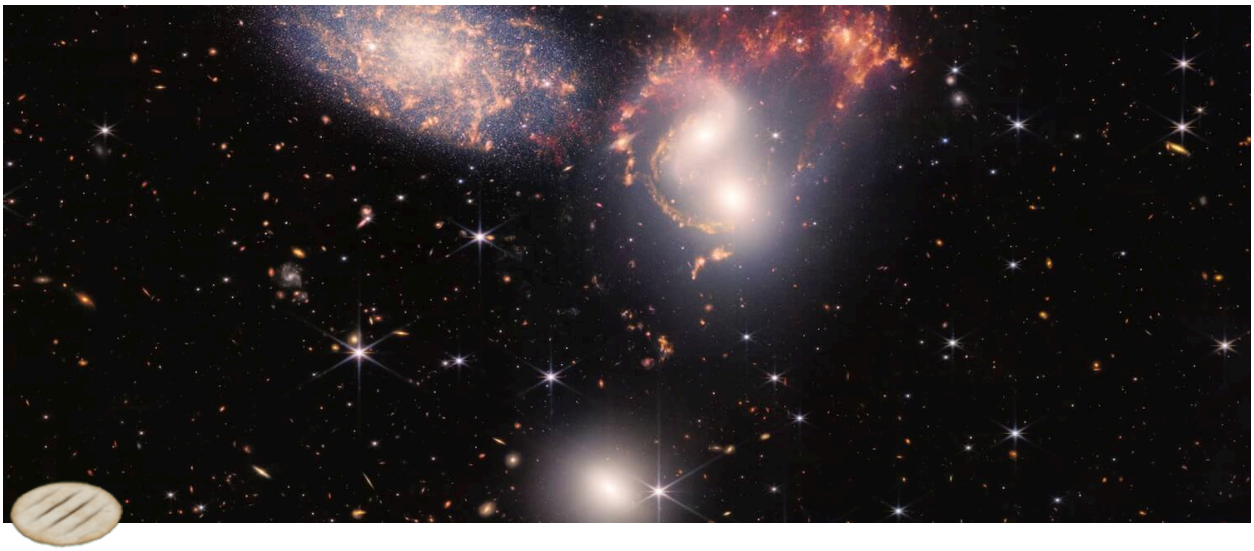

Backend Setup and API Usage Report

🛠️ Step-by-Step Setup Guide

1️⃣ Change directory

```
cd backend
```

2️⃣ Install Dependencies

Make sure you have **Node.js (v18+)** installed.

Then run:

```
npm install
```

3️⃣ Environment Variables

Create a `.env` file in the **root directory** with the following content:

```
DATABASE_URL="postgresql://<username>:<password>@localhost:5432/<dbname>?schema=public"
JWT_SECRET="yoursecretkey"
CLOUDINARY_URL="cloudinary://<api_key>:<api_secret>@<cloud_name>"
PORT=5000
```

> ⚠️ Tip: Never push your .env file to GitHub. Add it to .gitignore.

4️⃣ Prisma Setup

1. Generate Prisma Client:

   ```
   npx prisma generate
   ```

2. Push Database Schema to PostgreSQL:

   ```
   npx prisma db push
   ```

5️⃣ Cloudinary Setup

1. Create an account at https://cloudinary.com.
2. Go to **Dashboard → Account Details**.
3. Copy your **API Key, API Secret, and Cloud Name**.
4. Replace them inside `.env` in this format:

   ```
   CLOUDINARY_URL=cloudinary://<api_key>:<api_secret>@<cloud_name>
   ```

5. To test connection:

```
node cloudinaryTest.js
```

✅ If you see “Cloudinary connected successfully”, setup is complete.

6️⃣ Start the Server

Once everything is ready, run:

```
npm run dev
```

Server should start on:

👉 **http://localhost:5000**

🔗 API Endpoints

All routes are prefixed with

`http://localhost:5000/api`

1️⃣ Register User

POST `/auth/signup`

🧾 Request Body

```
{
  "firstName": "Alice",
  "email": "alice@example.com",
  "password": "123456",
  "birthday": "2000-05-12",
  "gender": "Female",
  "interestedIn": "Men",
  "lookingFor": "Relationship",
```

```
  "phoneNumber": "9876543211",
  "interests": ["Music", "Travel"],
  "sexualOrientation": "Straight",
  "profileImage": "https://dummyimage.com/200×200/000/fff.png&text=Profil
e"
}
```

✅ Sample Response

```
{
  "success": true,
  "message": "User registered successfully. OTP sent to email.",
  "user": {
    "id": "b4d3e7f5-79f9-4cb2-8bcd-d0c8a3421e67",
    "firstName": "Alice",
    "email": "alice@example.com",
    "isVerified": false}
}
```

2️⃣ Verify OTP

POST `/auth/verify-otp`

🧾 Request Body

```
{
  "email": "alice@example.com",
  "otp": "123456"
}
```

✅ Sample Response

```
{
  "success": true,
  "message": "User verified successfully"
}
```

3️⃣ Login User

POST `/auth/login`

🧾 Request Body


```
{
  "email": "alice@example.com",
  "password": "123456"
}
```


✅ Sample Response


```
{
  "success": true,
  "message": "Login successful",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "id": "b4d3e7f5-79f9-4cb2-8bcd-d0c8a3421e67",
    "firstName": "Alice",
    "email": "alice@example.com",
    "profileImage": "https://dummyimage.com/200×200/000/fff.png&text=Profi
le"
  }
}
```