Steps to run project locally:

1. Unzip the file to a single folder.
2. Open this folder in vscode or your favorite editor.
3. Run the following command to install requires libraries/dependencies : npm i
4. Change the name of the ".env copy" file to .env
5. Now you need to host the database somewhere and for that I have used supabase which provides free POSTGRES database. Create a project at their website and get connection string at Settings->Database Tab-> Database settings.
6. Make sure the port number is correct in your connection string and copy this url to DATABASE_URL.
7. Also set SECRET_KEY, you can use any string.
8. Now run the following commands:

   npx prisma migrate dev – This will create tables in supabase, and you can name this migration anything.

   npx prisma generate – This will generate prisma client code to interact with node.js.
9. Now run the command – nodemon index.js - to start the server.
10. You can test the endpoints using postman.

Endpoints

- **'/login'** - method POST – This endpoint is used to authenticate users.

  Fields **required** - phoneNumber, password

- **'/register'** - method POST – Create new user.

  Fields **required** – name, password, phoneNumber

  Optional fields – email

- **‘/user/:phoneNumber** - method GET -
  Enter the number you want to search in url and make sure when you login or register, you copy the token from the screen which is used for further authorization.

  In postman -> Authorization tab -> Select type as Bearer token and copy paste the token there.

  This will fetch the user details.

- **‘users/:name’ –** This endpoint generates a list of users sorted, the records who name starts with parameter are shown first and rest later.

  Required - Provide token in authorization header.

- **‘/reportSpam/:phoneNumber’ –** Reports a number as spam.

  Required – Provide token in authorization header.

Tech Stack Used:

- Node.js - For setting up server and logic.
- Prisma – ORM tool
- PostgresSQL – Database
- For Authentication/ Authorization – JSON web tokens (JWT)