[HOME] Home Rent & Meal Cost Management App

[SETUP] Technology Stack

Frontend: Angular + Angular Material

Backend: Express.js (Node.js)

Database: MongoDB (local instance)

Deployment: Serve on local PC and expose via ngrok

[STRUCTURE] Folder Structure

- Frontend: home-management-frontend/

- Backend: home-management-backend/

[BACKEND] Backend Setup (Express + MongoDB)

---

1. Create project & install dependencies:

   $ mkdir home-management-backend

   $ cd home-management-backend

   $ npm init -y

   $ npm install express mongoose cors dotenv

2. Create server.js and .env

   server.js connects to MongoDB and sets up Express server

3. MongoDB URI in .env:

   MONGO_URI=mongodb://127.0.0.1:27017/homeManagement

4. Run backend:

$ node server.js

[FRONTEND] Frontend Setup (Angular)

---

1. Create Angular project:

$ ng new home-management-frontend

$ cd home-management-frontend

$ ng add @angular/material

2. Setup HTTP proxy (proxy.conf.json):

```
{
  "/api": {
    "target": "http://localhost:5000",
    "secure": false,
    "changeOrigin": true
  }
}
```

3. Update start script in package.json:

"start": "ng serve --proxy-config proxy.conf.json --host 0.0.0.0"

4. Run Angular app:

$ ng serve --proxy-config proxy.conf.json --host 0.0.0.0

[NGROK] Use ngrok to Expose App Publicly

---

1. Install and run ngrok:

   $ ngrok http 4200

2. Optional: expose backend too

   $ ngrok http 5000

[NOTE] Notes

- IndexedDB/LocalStorage is device-specific.
- To enable access from any device, backend must handle all data.
- JSON export/import possible if frontend-only.

[SETUP] Next Steps: Feature Building

- Add members (name, room)
- Setup house (rooms, rent %)
- Add wallet entries per member
- Add daily bazar costs
- Add daily meal entries per member
- Calculate meal rate and balance

[DONE] Summary

---

1. Build backend and test on localhost:5000
2. Build Angular frontend and connect to backend
3. Serve both locally

4. Use ngrok to share publicly