



# Week 2 Task Plan – Backend Developer

**Project:** Household Services Booking Website

**Role:** Backend Developer

**Week:** 2 (Monday to Friday)

**Collaborating With:** Frontend Developer, UI/UX Designer

**Goal:** Set up backend structure, database, and build basic APIs (authentication + services)

---



## Project Overview

You are building the backend for a web platform where users can:

- Sign up and log in
- View available home services like AC repair, electrician, house cleaning, etc.
- Book services (booking feature will be done in Week 3)

The frontend developer is building the pages this week. Your job is to prepare the APIs that will connect to those pages starting next week.

---



## What You Will Do This Week

You will:

1. Set up the backend project and database
2. Create user and service models

3. Build API for user registration and login
  4. Create API to list all available services
  5. Add dummy service data
  6. Test APIs and share documentation
- 



## Tools You Can Use (Your Choice)

Use **any backend technology you are comfortable with**, such as:

- Node.js (Express), Django, Flask, Laravel, Spring Boot
  - MySQL, MongoDB, PostgreSQL, SQLite
  - Postman for testing your APIs
  - GitHub for sharing your code (recommended)
- 



## Day-by-Day Task Breakdown

---



### Day 1 – Project Setup

- Create a new backend project folder
- Set up basic project structure
  - Example folders: `routes/`, `controllers/`, `models/`, `config/`

- Install and configure basic packages (e.g., express, dotenv, body-parser if using Node.js)
- Add a test route:
  - Example: `GET /api/health` should return "Server is running"

✓ **Goal:** Your server runs on localhost and returns a simple test response.

---

## ✓ Day 2 – Database & Models

- Choose your database (MySQL, MongoDB, etc.)
- Connect it to your backend project
- Create 3 models/tables:
  1. **User:** name, email, password, role
  2. **Service:** name, description, price
  3. **Booking** (structure only for now): userId, serviceId, date, status

✓ **Goal:** Database is connected and models are ready.

---

## ✓ Day 3 – User Authentication APIs

- Create **User Registration API**
  - `POST /api/register`
  - Accepts name, email, password
  - Stores hashed password

- Create **Login API**
  - `POST /api/login`
  - Returns success response and token (JWT or session)

✓ **Goal:** You can register and log in a user using Postman.

---

#### ✓ **Day 4 – Services API + Dummy Data**

- Create **Get All Services API**
  - `GET /api/services`
  - Returns a list of services from the database
- (Optional) Create **Add Service API**
  - `POST /api/services` — can be used later by Admin
- Add 5–10 dummy services to your database (e.g., "AC Repair", "Electrician", etc.)

✓ **Goal:** Frontend developer can fetch services from your API in Week 3.

---

#### ✓ **Day 5 – Testing & Documentation**

- Test all APIs using Postman (register, login, services)
- Write basic API documentation:
  - Endpoint URL

- Method (GET/POST)
- What parameters are required
- Example request/response
- Push your code to GitHub
- Share your API base URL (e.g., <http://localhost:5000>) and documentation with the frontend developer

✓ **Goal:** APIs are working, tested, and shared with the team.

---

## ✓ APIs You Will Create This Week

Endpoint	Method	Auth	Description
<a href="#">/api/health</a>	GET	✗	Test if server is running
<a href="#">/api/register</a>	POST	✗	Register a new user
<a href="#">/api/login</a>	POST	✗	Log in a user
<a href="#">/api/services</a>	GET	✗	Get all services
<a href="#">/api/services</a>	POST	✓	Add a new service (optional)

---

## End of Week Deliverables

- ☒ Running backend project (localhost)
- ☒ Connected database with users and services
- ☒ Auth and services APIs tested in Postman
- ☒ Dummy services added to DB
- ☒ API documentation (shared with frontend dev)
- ☒ Code pushed to GitHub