Smart Parking Availability System

IoT-Based Project Plan Presentation (PPP)

Student: Garvin Patel

Course: IoT Project

Status: PPP Ready

Problem Statement

- Students waste time driving through parking garages searching for spots
- No visibility into garage capacity before entering
- Results in:
    - Time loss
    - Increased congestion
    - Frustration for students

Project Goal

- Design an **IoT-based system** that:
  - Collects sensor data
  - Transmits data wirelessly
  - Processes data on a backend server
  - Displays parking availability to users

Project Context (IoT Skeleton)

This project uses a **provided IoT hardware skeleton**:

- **Arduino Nano** – main controller
- **ESP32-C6** – WiFi / BLE communication
- **TMP102 (I2C Temperature Sensor)** – representative sensor

> The TMP102 is used to simulate parking-related data for system design and data flow.

Why a Temperature Sensor?

- Course-provided sensor
- Used to:
    - Practice sensor communication
    - Build end-to-end IoT data flow
    - Simulate parking availability logic

> The focus is on **architecture and integration**, not real parking detection.

Proposed Solution

1. TMP102 collects sensor data
2. Arduino Nano reads and processes data
3. ESP32-C6 transmits data wirelessly
4. Backend server receives and processes data
5. Web interface displays availability status

Key Features

- Sensor data collection (TMP102)
- Microcontroller-based processing
- Wireless transmission (ESP32-C6)
- Backend data handling
- Web-based availability display

Technologies Used

Hardware

- Arduino Nano
- ESP32-C6
- TMP102 Temperature Sensor

Software

- Arduino C/C++
- Node.js + Express
- MongoDB
- React

System Architecture Overview

Data Flow:

TMP102 → Arduino Nano → ESP32-C6 → Backend Server → Web Interface

> Architecture diagram will be added during implementation.

Backend Overview

- REST API built with **Node.js + Express**
- Receives sensor data from ESP32-C6
- Stores data in **MongoDB**
- Processes logic for availability status

Frontend Overview

- Built using **React**
- Displays:
  - Parking availability status
  - System updates
- Simple UI focused on clarity

AI-Assisted Learning Topics

Topic 1: IoT Debugging & Hardware Integration

Using AI to:

- Debug sensor communication issues
- Understand I2C and serial data flow
- Troubleshoot ESP32 WiFi problems

AI-Assisted Learning Topics (Continued)

Topic 2: Backend & API Design

Using AI to:

- Design REST APIs
- Validate data flow from IoT devices
- Improve code structure and documentation

Project Plan & Milestones

Sprint 1

- Hardware understanding and setup
- Sensor communication testing
- Define system requirements

Sprint 2

- Wireless data transmission
- Backend API development
- Frontend UI creation

Sprint 3

- System integration
- End-to-end testing
- Final documentation

Progress Tracking & Metrics

- GitHub commits
- Sprint milestone completion
- Successful data transmission
- Integration checkpoints

Risks & Mitigation

Risks

- IoT hardware learning curve
- Wireless communication failures
- Time constraints

Mitigation

- Incremental testing
- Simulated data
- AI-assisted troubleshooting

PPP Readiness Statement

- System design fully documented
- Architecture and data flow defined
- Clear milestones and risks identified

This project is ready for PPP review and discussion.

Thank You

Questions & Feedback Welcome