ours available.
the tours.

usTrackingMachanism

OBJECTIVE

Week 1: Research and Setup

Goal: Establish the foundation.

Tasks:

Day 1-2: Research how GPS tracking works and explore APIs (e.g., Google Maps API). Review Py
Day 3: Create a basic project folder structure on your laptop (include folders for frontend, backe
Day 4-5: Set up the tools needed for the project (install Flask/Django, Python, HTML editor, an
Day 6-7: Review tutorials on creating a simple Flask web app with database integration.

Time Commitment: 1-2 hours/day, flexible during weekends.

Week 2: Basic Frontend Design

Goal: Build the user interface.

Tasks:

Day 1-3: Design wireframes for the bus tracking interface (e.g., input field for bus number, map
Day 4-7: Code the frontend using HTML, CSS, and JavaScript. Focus on creating:
- A clean and responsive design.
- A navigation bar and footer.
- A placeholder for displaying the map.

Time Commitment: 1-2 hours/day on weekdays, 4-5 hours on weekends.

Week 3: Backend Development

Goal: Set up the backend for handling user requests.

Tasks:

Day 1-2: Design a basic Flask/Django app structure.
Day 3-4: Code the API endpoints for:
- Fetching bus routes from the database.
- Handling GPS location updates.

Day 5-7: Connect the backend to the frontend (via AJAX or Fetch API).

Time Commitment: 1-2 hours/day on weekdays, 4-5 hours on weekends.

Week 4: Google Maps Integration

Goal: Display real-time bus locations and routes on the map.

Tasks:

Day 1-3: Explore Google Maps API (or alternatives like OpenStreetMap).
Day 4-5: Write Python code to fetch and display bus locations on the map.
Day 6-7: Test the integration and troubleshoot issues (e.g., incorrect locations or map loading e

Time Commitment: 1-2 hours/day on weekdays, 4-5 hours on weekends.

Week 5: GPS Data Simulation

Goal: Create mock GPS data for testing.

Tasks:

- **Day 1-3**: Write a script to simulate bus movement (generate random GPS coordinates or use rea
- **Day 4-5**: Test the simulated data with the map and tracking mechanism.
- **Day 6-7**: Refine the simulation and ensure smooth updates on the interface.

Time Commitment: 1-2 hours/day on weekdays, 4-5 hours on weekends.

Week 6: Database Implementation

Goal: Store bus routes, user data, and tracking history.

Tasks:

- **Day 1-3**: Design the database schema in MySQL (tables for bus routes, location updates, and use
- **Day 4-5**: Implement database integration with the backend (Python + MySQL).
- **Day 6-7**: Test CRUD operations (Create, Read, Update, Delete) for database entries.

Time Commitment: 1-2 hours/day on weekdays, 4-5 hours on weekends.

Week 7: Final Integration and Debugging

Goal: Connect all components and ensure smooth functionality.

Tasks:

- **Day 1-3**: Run end-to-end tests on the system:
  Submit bus number → View routes → Display real-time tracking on map.
- **Day 4-5**: Debug any issues in the frontend, backend, or database.
- **Day 6-7**: Optimize the code and improve interface responsiveness.

Time Commitment: 1-2 hours/day on weekdays, 5-6 hours on weekends.

Week 8: Polishing and Documentation

Goal: Prepare for presentation and submission.

Tasks:

- **Day 1-2**: Add finishing touches to the UI (e.g., loading animations, better styling).
- **Day 3-4**: Write project documentation:
  Project overview.
  How to run the system.
  Challenges faced and solutions implemented.
- **Day 5-7**: Create a presentation/demo for your teacher and team.

Time Commitment: 1-2 hours/day on weekdays, flexible on weekends.