**. Project Setup**

* **Project Name:** TrackLine
* **Tech Stack:**
  + **Backend:** Django
  + **Frontend:**Django
  + **Version Control:** GitHub

**2. User Stories & Features**

1. **User Authentication:**
   * Registration
   * Login/Logout
   * Forgot Password
2. **User Types:**
   * **Registered Users:**
     + Log rides and track CO2 emissions
     + View dashboard with interactive charts and graphs
     + Receive personalized recommendations for eco-friendly transportation
     + Upload files (photos, documents, IDs)
     + View and edit profile
   * **Unregistered Users:**
     + Access articles regarding CO2 awareness
     + Use search functionality with dropdown for refined searches
3. **Dashboard:**
   * Display cumulative CO2 emissions over time
   * Interactive charts and graphs for emission trends, peak periods, and comparative analyses
   * Suggest alternative transportation options
4. **Ride Logging:**
   * Log rides with vehicle type and distance
   * Calculate CO2 emissions based on input data
5. **Community Features:**
   * Share progress and tips
   * View community posts and achievements
6. **Additional Pages:**
   * Contact Us
   * About Us
   * Team Details
   * Footer with reference websites
7. **Data & Analytics:**
   * Track user visits and interactions using cookies and sessions
   * Display user history and visit statistics
8. **Administrative Functions:**
   * Manage user accounts
   * Monitor and manage site content

**3. Models**

* **Users:**
  + ID (Primary Key)
  + Username
  + Email
  + Password (hashed)
  + Profile Info (JSON)
  + Registration Date
  + Last Login Date
* **Rides Table:**
  + ID (Primary Key)
  + User ID (Foreign Key)
  + Vehicle Type
  + Distance
  + CO2 Emissions
  + Date Logged
* **Articles Table:**
  + ID (Primary Key)
  + Title
  + Content
  + Date Published
* **Community Posts Table:**
  + ID (Primary Key)
  + User ID (Foreign Key)
  + Content
  + Date Posted

**4. Wireframes & Mockups**

* **Homepage:**
  + Navigation Bar (Login, Register, Contact Us, About Us)
  + Hero Section (Promotional Text)
  + Search Bar
  + Articles Section (List of Articles)
* **Dashboard (Registered Users):**
  + Navigation Bar (Dashboard, Log Ride, Community, Profile)
  + CO2 Emissions Summary
  + Interactive Charts and Graphs
  + Recommendations Section
  + Community Updates
* **Ride Logging Page:**
  + Form to log ride details (Vehicle Type, Distance)
  + Submit Button
* **Profile Page:**
  + Profile Details
  + Uploaded Files
  + Edit Profile Button

**5. GitHub Repository Setup**

1. **Initialize Repository:**
   * Create a new repository named TrackLine on GitHub.
   * Clone the repository to local machine.
   * Initialize Django project and app.
2. **Directory Structure:**

TrackLine/

├── mysite/

│ ├── settings.py

│ ├── urls.py

│ ├── wsgi.py

│ └── ...

├── myapp/

│ ├── migrations/

│ ├── static/

│ ├── templates/

│ ├── admin.py

│ ├── apps.py

│ ├── models.py

│ ├── views.py

│ ├── urls.py

│ └── ...

├── manage.py

├── README.md

├── requirements.txt

└── .gitignore

**6. Development Workflow**

* **Branching Strategy:**
  + main branch for stable code
  + dev branch for development
  + Feature branches (feature/user-auth, feature/dashboard, etc.)
* **Task Allocation:**
  + **Backend:** Models, Views, URL Routing
  + **Frontend:** Templates, Static Files (CSS, JS), Bootstrap Integration
  + **Database:** PostgreSQL Setup, Migrations
  + **Testing:** Unit Tests, Integration Tests
* **Continuous Integration:**
  + Set up GitHub Actions for CI/CD
  + Run tests on pull requests
  + Deploy to staging environment on successful build

**7. Bootstrap Styling**

* **Global Styles:**
  + Include Bootstrap CSS and JS files in base.html.
  + Customize Bootstrap with a custom stylesheet (style.css).
* **Responsive Design:**
  + Use Bootstrap grid system for layout.
  + Ensure all pages are mobile-friendly.

**8. Interactive Charts & Graphs**

* Use Chart.js or D3.js for interactive charts.
* Display data such as CO2 emissions over time, peak emission periods, and comparative analyses.

**9. File Uploads**

* Use Django's built-in file upload capabilities.
* Store uploaded files securely and associate them with user profiles.

**10. Deployment**

* **Staging Environment:**
  + Deploy to Heroku or AWS for staging.
  + Set up PostgreSQL database.
* **Production Environment:**
  + Deploy to a scalable cloud platform.
  + Ensure security and scalability measures are in place.

**11. Documentation**

* Document code and features in README.md.
* Create user manuals and developer guides.
* Maintain changelogs and contribution guidelines.