# **Streamlit App Plan for Uber Eats Session Logger**

## **1. Project Overview**

Build an interactive Streamlit application to:

* Record and manage Uber Eats delivery sessions and trips.
* Visualize metrics like total earnings, average trip durations, and earnings per hour.
* Save and load data persistently (e.g., CSV or local file).

## **2. Functional Requirements**

### **2.1. User Inputs**

* **Session Entry Form**: Input fields for date, start time, trips count, shift length, earnings, device used.
* **Trip Entry Form**: Input fields for restaurant, duration, mileage, delivery zone, tip, earnings.

### **2.2. Data Management**

* Auto-calculate session end time from start time and shift length.
* Auto-generate session and trip IDs.
* Display current session and trip data in tabular format.
* Option to delete or modify entries.

### **2.3. File Operations**

* Save sessions and trips to local CSV files.
* Load from existing CSV files.
* Manual export/download feature.

### **2.4. Analytics & Visualization**

* Total earnings over time.
* Earnings per session/trip.
* Trip count per device or zone.
* Time-based plots (e.g., shift hours vs earnings).

## **3. Technical Requirements**

### **3.1. Stack**

* **Frontend**: Streamlit
* **Backend Logic**: Python (pandas, datetime)
* **Storage**: Local CSV files

### **3.2. Packages**

streamlit

pandas

numpy

matplotlib

## **4. App Layout & User Flow**

### **Page 1: Data Entry**

* Tabs: "Add Session", "Add Trip"
* After submission, show updated tables.

### **Page 2: Session & Trip Tables**

* Paginated display
* Sortable, filterable by date or device

### **Page 3: Visualizations**

* Plots for earnings, mileage, durations
* Filter by date range, device

### **Page 4: Settings**

* Upload/download CSV
* Reset all data

## **5. Backend Logic**

* Helper functions for data parsing and ID generation.
* Logic to handle datetime parsing and duration formatting.
* Validation for user inputs (e.g., positive numbers, time format).

## **6. UI Components**

* st.text\_input, st.date\_input, st.number\_input, st.selectbox
* st.dataframe or st.table for data views
* st.pyplot or st.altair\_chart for visualizations

## **7. Next Steps**

1. Convert current notebook functions into Python modules.
2. Design Streamlit layout using tabs/pages.
3. Implement forms and display logic.
4. Add persistent data handling.
5. Add visualizations and export features.
6. Test and deploy locally.