APPENDIX (SAMPLE CODE)

APPENDIX 1:

FRONTEND:

LOGIN PAGE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Traffic Prediction Management - Login</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }
        .header {
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            text-align: center;
        }
        .container {
            max-width: 400px;
            margin: 100px auto;
            background: #fff;
            padding: 20px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }
        .form-group {
            margin-bottom: 15px;
```

```
}
label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}
input[type="text"],
input[type="password"] {
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
}
button {
    background-color: #333;
    color: #fff;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    width: 100%;
}
button:hover {
    background-color: #555;
}
.link-buttons {
    display: flex;
    justify-content: space-between;
    margin-top: 20px;
}
.link-buttons a {
    text-decoration: none;
    color: white;
    padding: 10px 15px;
```

```
        background-color: #333;
        border-radius: 5px;
        text-align: center;
    }
    .link-buttons a:hover {
        background-color: #555;
    }

    /* Popup styles */
    .popup {
        display: none;
        position: fixed;
        left: 50%;
        top: 50%;
        transform: translate(-50%, -50%);
        background-color: rgba(0, 0, 0, 0.7);
        color: #fff;
        padding: 20px;
        border-radius: 10px;
        text-align: center;
    }
    .popup button {
        background-color: #4CAF50;
        border: none;
        padding: 10px 20px;
        color: white;
        border-radius: 5px;
        cursor: pointer;
        margin-top: 10px;
    }
    .popup button:hover {
        background-color: #45a049;
    }
</style>
```

```
    <script>
        function login() {
            // Display the pop-up message
            document.getElementById('login-popup').style.display = 'block';

            // After a short delay, redirect to the User page
            setTimeout(function() {
                window.location.href = "User.html"; // Redirect to User.html after 2
seconds
            }, 2000);  // 2000ms = 2 seconds
        }
    </script>
</head>
<body>
    <div class="header">
        <h1>Traffic Prediction Management</h1>
    </div>
    <div class="container">
        <div class="form-group">
            <label for="username">Username:</label>
            <input type="text" id="username" placeholder="Enter username">
        </div>
        <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" id="password" placeholder="Enter password">
        </div>
        <button onclick="login()">Login</button>
        <div class="link-buttons">
            <a href="settings.html">Settings</a>
        </div>
    </div>

    <!-- Pop-up message -->
    <div id="login-popup" class="popup">
```

```
        <p>Login Successful!</p>
        <button>Redirecting...</button>
    </div>
</body>
</html>
```

USER:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Traffic Prediction Management</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }
        .header {
            position: absolute;
            top: 0;
            width: 100%;
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            text-align: center;
        }
```

```
        .container {
            text-align: center;
        }
        button {
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            width: 200px;
            margin: 10px;
            font-size: 16px;
        }
        button:hover {
            background-color: #555;
        }
    </style>
    <script>
        function goToLiveTraffic() {
            window.location.href = "live_traffic.html"; // Redirect to the Live Traffic
page
        }

        function goToLiveViolation() {
            window.location.href = "live_violation.html"; // Redirect to the Live
Violation page
        }
    </script>
</head>
<body>
    <div class="header">
        <h1>Traffic Prediction Management</h1>
    </div>
```

```
    <div class="container">
        <button onclick="goToLiveTraffic()">Live Traffic</button>
        <button onclick="goToLiveViolation()">Live Violation</button>
    </div>
</body>
</html>
```

LIVE TRAFFIC:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Live Traffic</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }
        button {
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
```

```
        }

        button:hover
 {
            background-color: #555;
        }
    </style>
    <script>
        function watchLiveTraffic() {
            alert("Redirecting to Live Traffic Stream...");
            // Replace with actual live traffic stream URL
            window.location.href = "live_traffic_stream.html";
        }
    </script>
</head>
<body>
    <button onclick="watchLiveTraffic()">Click to Watch Live Traffic</button>
</body>
</html>
```

LIVE VIOLATION:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Live Violation</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
```

```
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }
        button {
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
        }
        button:hover {
            background-color: #555;
        }
    </style>
    <script>
        function watchLiveViolation() {
            alert("Redirecting to Live Violation Stream...");
            // Replace with actual live violation stream URL
            window.location.href = "live_violation_stream.html";
        }
    </script>
</head>
<body>
    <button onclick="watchLiveViolation()">Click to Watch Live Violation</button>
</body>
</html>
```

SETTINGS:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Settings</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }
        .header {
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            text-align: center;
        }
        .container {
            max-width: 600px;
            margin: 50px auto;
            background: #fff;
            padding: 20px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }
        .form-group {
            margin-bottom: 15px;
        }
        label {
            display: block;
            font-weight: bold;
            margin-bottom: 5px;
        }
        input[type="text"] {
```

```
            width: 100%;
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }
        button {
            background-color: #333;
            color: #fff;
            padding: 10px 15px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            width: 100%;
        }
        button:hover {
            background-color: #555;
        }
    </style>
    <script>
        function saveSettings() {
            alert("Settings Saved!"); // Display pop-up message
        }
    </script>
</head>
<body>
    <div class="header">
        <h1>Settings</h1>
    </div>
    <div class="container">
        <h3>Update Your Settings</h3>
        <div class="form-group">
            <label for="username">Update Username:</label>
            <input type="text" id="username" placeholder="New username">
        </div>
```

```
        <div class="form-group">
            <label for="email">Update Email:</label>
            <input type="text" id="email" placeholder="New email">
        </div>
        <button onclick="saveSettings()">Save Changes</button> <!-- Save button
 triggers alert -->
    </div>
</body>
</html>
```

BACKEND:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# Load the data
file_path = 'Traffic.csv'  # Update with the correct path if needed
traffic_data = pd.read_csv(file_path)

# Convert Time to a usable feature (hour of the day)
traffic_data['Hour'] = pd.to_datetime(traffic_data['Time'], format='%I:%M:%S %p').dt.hour

# Part 1: Traffic Prediction
X = traffic_data[['Hour', 'CarCount', 'BikeCount', 'BusCount', 'TruckCount']]
y = traffic_data['Total']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest Regressor
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train, y_train)
```

```
# Predict on test data
traffic_data['Predicted_Total'] = rf_model.predict(X)

# Visualize Predictions
plt.figure(figsize=(14, 7))
plt.plot(traffic_data['Time'][:500], traffic_data['Total'][:500], label='Actual Traffic',
color='blue', marker='o', linewidth=1)
plt.plot(traffic_data['Time'][:500], traffic_data['Predicted_Total'][:500], label='Predicted
Traffic', color='orange', linestyle='--', linewidth=1)
plt.xticks(rotation=90, fontsize=8)
plt.title('Traffic Prediction vs Actual', fontsize=16)
plt.xlabel('Time (First 500 Observations)', fontsize=14)
plt.ylabel('Number of Vehicles', fontsize=14)
plt.legend(fontsize=12)
plt.grid(axis='y', linestyle='--', linewidth=0.7)
plt.tight_layout()
plt.show()

# Print model evaluation
mse = mean_squared_error(y_test, rf_model.predict(X_test))
print(f"Model Performance: Mean Squared Error = {mse:.2f} vehicles.")

# Part 2: Traffic Violation Analysis
traffic_situation_counts = traffic_data['Traffic Situation'].value_counts()

# Pie chart for Traffic Situations
plt.figure(figsize=(8, 8))
traffic_situation_counts.plot.pie(
    autopct='%1.1f%%',
    startangle=90,
    colors=sns.color_palette('pastel'),
    wedgeprops={'linewidth': 1, 'edgecolor': 'black'}
)
plt.title('Distribution of Traffic Situations', fontsize=16)
plt.ylabel('')
plt.show()

# Bar chart for High Traffic Days
traffic_data['High Traffic'] = traffic_data['Total'] > traffic_data['Total'].mean()
high_traffic_days = traffic_data.groupby('Day of the week')['High Traffic'].sum()
```

```
plt.figure(figsize=(10, 6))
high_traffic_days.plot(kind='bar', color='teal', edgecolor='black')
plt.title('High Traffic Days (Total Above Average)', fontsize=16)
plt.xlabel('Day of the Week', fontsize=14)
plt.ylabel('Number of High Traffic Periods', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', linewidth=0.7)
plt.tight_layout()
plt.show()

# Correlation Heatmap for Vehicle Counts
correlation_matrix = traffic_data[['CarCount', 'BikeCount', 'BusCount', 'TruckCount',
'Total']].corr()

plt.figure(figsize=(10, 7))
sns.heatmap(
    correlation_matrix,
    annot=True,
    cmap='Blues',
    fmt='.2f',
    linewidths=1,
    linecolor='black'
)
plt.title('How Different Vehicle Counts Relate to Total Traffic', fontsize=16)
plt.show()

# Part 3: 14 Plots for All Seven Days

# Chart 1 and Chart 2 for each day
unique_days = traffic_data['Day of the week'].unique()

for day in unique_days:
    # Filter data for the current day
    day_data = traffic_data[traffic_data['Day of the week'] == day]

    # Chart 1: Vehicle counts by hour
    hourly_vehicle_counts = day_data.groupby('Hour')[['CarCount', 'BikeCount', 'BusCount',
'TruckCount']].sum()
    plt.figure(figsize=(12, 6))
```

```
    hourly_vehicle_counts.plot(kind='bar', stacked=True, figsize=(12, 6), color=['blue',
'green', 'orange', 'red'], edgecolor='black')
    plt.title(f"Vehicle Counts by Hour: {day}", fontsize=16)
    plt.xlabel("Hour of the Day", fontsize=14)
    plt.ylabel("Number of Vehicles", fontsize=14)
    plt.xticks(rotation=0)
    plt.legend(title="Vehicle Type", fontsize=12)
    plt.grid(axis='y', linestyle='--', linewidth=0.7)
    plt.tight_layout()
    plt.show()

    # Chart 2: Average traffic per hour
    hourly_avg_traffic = day_data.groupby('Hour')['Total'].mean()
    plt.figure(figsize=(12, 6))
    hourly_avg_traffic.plot(kind='line', marker='o', linestyle='-', color='purple', linewidth=2)
    plt.title(f"Average Traffic per Hour: {day}", fontsize=16)
    plt.xlabel("Hour of the Day", fontsize=14)
    plt.ylabel("Average Total Traffic", fontsize=14)
    plt.grid(axis='both', linestyle='--', linewidth=0.7)
    plt.tight_layout()
    plt.show()
# Chart 1: Vehicle counts by hour for each day
unique_days = traffic_data['Day of the week'].unique()

for day in unique_days:
    day_data = traffic_data[traffic_data['Day of the week'] == day]
    hourly_vehicle_counts = day_data.groupby('Hour')[['CarCount', 'BikeCount', 'BusCount',
'TruckCount']].sum()

    plt.figure(figsize=(12, 6))
    hourly_vehicle_counts.plot(kind='bar', stacked=True, figsize=(12, 6), color=['blue',
'green', 'orange', 'red'], edgecolor='black')
    plt.title(f"Vehicle Counts by Hour: {day}", fontsize=16)
    plt.xlabel("Hour of the Day", fontsize=14)
    plt.ylabel("Number of Vehicles", fontsize=14)
    plt.xticks(rotation=0)
    plt.legend(title="Vehicle Type", fontsize=12)
    plt.grid(axis='y', linestyle='--', linewidth=0.7)
    plt.tight_layout()
    plt.show()
```

```
# Chart 2: Average traffic per hour for each day
for day in unique_days:
    day_data = traffic_data[traffic_data['Day of the week'] == day]
    hourly_avg_traffic = day_data.groupby('Hour')['Total'].mean()

    plt.figure(figsize=(12, 6))
    hourly_avg_traffic.plot(kind='line', marker='o', linestyle='-', color='purple', linewidth=2)
    plt.title(f"Average Traffic per Hour: {day}", fontsize=16)
    plt.xlabel("Hour of the Day", fontsize=14)
    plt.ylabel("Average Total Traffic", fontsize=14)
    plt.grid(axis='both', linestyle='--', linewidth=0.7)
    plt.tight_layout()
    plt.show()
```