UK Train Rides – Data Analytics & Dashboard Project

1. Project Planning & Management

1.1 Project Proposal

Objective

This project analyzes the operational performance of a real railway network using more than 31,000 actual booking transactions.
The main objectives are to:

- Discover booking and purchasing patterns.
- Evaluate trip performance (punctuality, delays, and cancellations).
- Identify operational issues such as frequent delays and refund requests.
- Understand passenger purchasing and payment behaviors.

Using Python for data preparation and analysis, and Power BI for visualization, the project aims to build an integrated analytical dashboard that supports decision-making and highlights improvement opportunities in the railway system.

Scope

- **Booking Patterns**:
    - Analyze reservation frequency over time.
    - Identify seasonal peaks and off-peak periods.
    - Detect the most popular routes and stations.
- **Trip Performance**:
    - Measure punctuality and delays.
    - Monitor cancellations and journey status.
    - Compare performance across routes, regions, and time periods.

- **Refunds & Complaints**:
    - Analyze refund request frequency.
    - Identify routes, time periods, or delay reasons associated with higher refund rates.
- **Payment Behavior**:
    - Study distribution of payment methods (Card, Online, Cash, etc.).
    - Understand how customers prefer to purchase tickets (timing, method, price ranges).
- **Operational Insights**:
    - Highlight problematic routes and time windows.
    - Support optimization of schedules and resource allocation.
    - Provide insights to improve customer satisfaction and service reliability.

1.2 Project Plan

Phase	Tasks	Duration	Milestone
1 – Data Cleaning & Transformation	Handle missing values, fix inconsistencies, unify formats, and prepare the dataset for analysis using Python (Pandas).	Week 2	Cleaned dataset ready
2 – Data Analysis & Metric Creation	Analyze patterns of delays, refunds, payments, booking volumes, and trip efficiency. Create calculated metrics.	Week 3	Key insights identified

Phase	Tasks	Duration	Milestone
3 – Dashboard Development	Build an interactive Power BI dashboard including KPIs, charts, and drill-down reports.	Week 4	First version of dashboard
4 – Review & Refinement	Validate numbers, review visuals, and enhance performance and usability of the dashboard.	Week 5	Final dashboard ready
5 – Presentation & Documentation	Prepare final presentation, project report, and publish the work on GitHub or similar platform.	Week 6	Project completed

---

1.3 Resources

- **Tools**
  - Python (for data cleaning, transformation, and exploratory analysis)
  - Power BI (for interactive dashboards and visualizations)
  - Excel / CSV (for initial data storage and export)
- **Python Libraries**
  - Pandas – data cleaning, transformation, and aggregation
  - Matplotlib and Seaborn – exploratory and diagnostic visualizations
- **Data Source**
  - DEPI Datasets – *UK Train Rides* dataset (over 31,000 transactions)

1.4 Task Assignment & Roles

(To be filled by the team)

Task	Team Member
Data Cleaning & Transformation	Mohammed Al-Saeed
Data Analysis & Metrics Creation	Ali Mohammed
Dashboard Development	Abdelrahman Ahmed
Review & Refinement	Mazen Ezzat
Presentation & Documentation	Mazen Ezzat

1.5 Risk Assessment & Mitigation

Risk	Mitigation Strategy
Missing or inconsistent data	Use validation rules and Python scripts to handle missing values, outliers, and inconsistent formats. Document any assumptions.
Performance issues in Power BI	Optimize data model (Star Schema), use aggregated tables when needed, and apply efficient DAX measures.
Misinterpretation of insights	Cross-check results with Python outputs and, when possible, validate with domain experts or supervisors.
Scope creep	Stick to clearly defined KPIs and use cases; additional ideas can be documented as future work.

1.6 Key Performance Indicators (KPIs)

- Total number of trips.

- On-time performance ratio (on-time journeys / total journeys).
- Average delay time per route / per station.
- Total number of refund requests and refund rate.
- Payment method distribution.
- Top 10 routes by booking volume.
- Trips by region or station (e.g., per departure station).
- Customer satisfaction score (derived metric based on delay, refund, and journey status).

1.7 Dataset Overview

Column Name	**Description**
Transaction ID	Unique identifier for each booking transaction
Trip ID	Identifier for each train journey
Departure Time	Scheduled departure time
Arrival Time	Scheduled arrival time
Actual Arrival Time	Actual arrival time of the train (if available)
Delay (minutes)	Delay duration in minutes
Journey Status	Final status of the journey (On time, Delayed, Cancelled, etc.)
Refund Request	Indicator if a refund was requested (Yes/No)
Payment Method	Type of payment used (Card, Online, Cash, etc.)

Column Name	Description
Ticket Price	Total fare paid by the passenger
Ticket Class	Class of service (e.g., Standard, First Class)
Ticket Type	Type of ticket (Single, Return, Season, etc.)
Route	Route or line name
Station From	Departure station
Station To	Arrival station
Railcard	Railcard type (if used)
Customer ID	Passenger identifier
Date of Travel	Actual travel date
Booking Date	Date when the ticket was purchased
Time of Purchase	Time when the ticket was purchased
Reason for Delay	Categorical description for delay cause (if available)

---

2. Literature Review

This project is inspired by common practices in transport analytics and railway performance monitoring, where large-scale ticketing and journey data are used to:

- Measure punctuality, delays, and reliability of services.
- Analyze demand patterns to optimize scheduling and capacity.
- Understand refund and compensation trends to reduce financial losses.
- Improve the passenger experience through data-driven decisions.

Feedback from railway analysts, academic mentors, or supervisors will be used to:

- Validate whether the KPIs and metrics are relevant and correctly defined.
- Suggest additional analyses such as forecasting or route-level benchmarking.
- Improve dashboard design, interactivity, and clarity.

Possible enhancements include:

- Adding **trend forecasting visuals** (e.g., future delays or demand using time-series models).
- Including **what-if analysis** (e.g., impact of delay reduction on refunds and satisfaction).

Final grading and evaluation criteria may be based on:

- Data quality and correctness of transformations.
- Relevance and accuracy of KPIs.
- Clarity, interactivity, and usability of the dashboard.
- Ability to support decision-making and operational improvements.

---

3. Requirements Gathering

3.1 Stakeholder Analysis

- **Business Managers / Operations Managers**
    - Focus on operational efficiency, punctuality, and revenue.
    - Need clear KPIs to identify underperforming routes or time periods.
- **Transport Authorities / Regulators**

    - Require reports on delays, cancellations, and service reliability.
    - Interested in long-term performance trends.
- **Passengers (Indirect Stakeholders)**
    - Benefit from improved reliability, clearer schedules, and reduced delays.

3.2 User Stories

- *"As an operations manager, I want to monitor train delays so that I can identify problematic routes and time periods and take corrective actions."*
- *"As a financial analyst, I want to track refund trends and their relations to delays and routes so that I can understand and reduce operational losses."*
- *"As a business manager, I want to see booking volumes per route and per region so I can optimize capacity and pricing."*

3.3 Functional Requirements

- An interactive dashboard showing key KPIs:
    - Delays, cancellations, refunds, bookings, and revenue.
- Filters (slicers) by:
    - Date / Time period.
    - Route / Station / Region.
    - Payment method.
    - Ticket class / Ticket type.
    - Railcard usage.
- Visuals such as:
    - Time-series charts for bookings and delays.

    - Bar charts for top routes and stations.
    - Distribution charts for payment methods and ticket classes.
    - Maps (if geographical data is available) for regional performance.
- Ability to drill down from high-level KPIs to detailed transaction-level records.

3.4 Non-Functional Requirements

- Dashboard should load in **≤ 5 seconds** under normal usage.
- Clear, consistent, and accessible visualization design:
    - Professional color palette (e.g., blue/grey tones).
    - Legible fonts and labels.
- Scalable data model:
    - Ability to add future datasets or additional years without redesigning everything.
- Maintainability:
    - Python scripts and Power BI files should be clearly organized, commented, and version-controlled (e.g., via GitHub).

---

4. System Analysis & Design

4.1 Problem Statement

Manual analysis of train operations using raw CSV files is time-consuming and prone to error.
It is difficult to manually detect patterns in delays, refunds, and booking behaviors across tens of thousands of records.

This project automates the analytics process using:

- Python for data cleaning, transformation, and calculations.

- Power BI for interactive visualization and reporting.

The result is a decision-support dashboard that helps stakeholders quickly understand performance and take data-driven actions.

4.2 Use Case Overview

- **Admin**
    - Updates and maintains the dataset (e.g., uploads new CSVs).
- **Data Analyst**
    - Cleans and prepares the data using Python.
    - Defines KPIs and calculated metrics.
- **Business User / Manager**
    - Interacts with the Power BI dashboard.
    - Applies filters and reads KPIs to support decisions.

4.3 Software Architecture

- **Data Source**
    - Raw CSV/Excel files (UK Train Rides dataset from DEPI).
- **Backend Processing (ETL Layer)**
    - Python scripts:
        - Load raw data.
        - Clean missing or inconsistent values.
        - Create derived columns (e.g., delay flags, on-time status, weekend/weekday).
        - Export a clean, modeled dataset ready for Power BI.

- **Visualization Layer**
  - Power BI:
    - Implements the star schema (Fact & Dimension tables).
    - Builds KPIs, DAX measures, and visuals.
    - Provides interactive filters and drill-down capabilities.
- **Data Flow**
  - CSV → Python (Cleaning & Transformation) → Exported Clean Data (CSV / Excel) → Power BI → Dashboard View.

4.4 Data Model – ER Diagram & Star Schema

The central table is **Fact_Train_Rides**, connected to multiple dimension tables:

- **Fact_Train_Rides**
  (Transaction_ID, Departure_Station_ID, Arrival_Station_ID, Journey_Status_ID,
  Purchase_Date_ID, Journey_Date_ID, Purchase_Time_ID, Departure_Time_ID,
  Arrival_Time_ID, Actual_Arrival_Time_ID, Ticket_ID, Payment_ID, Railcard_ID,
  DelayReason_ID, Refund_Request, Price)
- **Dim_Station** (Station_ID, Station_Name, Region)
- **Dim_Date** (Date_ID, Full_Date, Day, Month, Month_Name, Quarter, Year, Weekday_Name, Is_Weekend)
- **Dim_Time** (Time_ID, Full_Time, Hour, Minute, Time_of_Day)
- **Dim_Payment** (Payment_ID, Payment_Method)
- **Dim_Ticket** (Ticket_ID, Ticket_Class, Ticket_Type)

- **Dim_Railcard** (Railcard_ID, Railcard_Name)
- **Dim_Status** (Journey_Status_ID, Journey_Status)
- **Dim_DelayReason** (DelayReason_ID, Reason_For_Delay)

This star schema design supports fast aggregations on trips, delays, refunds, and revenue across different dimensions (station, date, time, ticket type, payment method, etc.).

4.5 Physical Schema

- **Primary Keys**
  - Transaction_ID on Fact_Train_Rides.
  - Station_ID, Date_ID, Time_ID, Payment_ID, Ticket_ID, Railcard_ID, Journey_Status_ID, DelayReason_ID on dimension tables.
- **Relationships**
  - Each foreign key in Fact_Train_Rides links to the corresponding dimension key.
  - This provides a unified model for integrated analysis of:
    - Journey performance (Status, DelayReason).
    - Ticket and payment behavior (Ticket, Payment, Railcard).
    - Time and date patterns (Date, Time).
    - Spatial information (Stations, Regions).

4.6 Data Flow & System Behavior

- **Data Flow Diagram (High-Level)**
  1. Extract raw data from DEPI dataset (CSV).

2. Transform and clean the data using Python scripts.
3. Load transformed data into Power BI.
4. Build KPIs and visuals and publish dashboard.

- **Activity Flow**
    - Load raw files → Validate and clean → Create derived columns → Export cleaned dataset →
      Import to Power BI → Define relationships and measures → Design visuals → Publish and review.

4.7 UI/UX Design & Prototyping

- **Wireframes**
    - Main dashboard page with:
        - KPI cards (Total Trips, On-time %, Avg Delay, Refunds, Revenue).
        - Charts for delays, bookings, and payment distribution.
        - Filters on the left or top (Date, Route, Station, Payment Method, Ticket Class).
- **Design Guidelines**
    - Use professional, calm color palette (blue/grey tones).
    - Clear typography and readable axis labels.
    - Interactive slicers and drill-downs to allow users to explore data.
    - Avoid clutter; each page should answer a specific set of questions.

---

5. Deployment System & Integration

- **Technology Stack**
    - Python (Pandas, Matplotlib, Seaborn).
    - Power BI Desktop and Power BI Service.
    - CSV/Excel as intermediate data storage.
- **Deployment Diagram (Conceptual)**

1. Data stored locally as CSVs.
2. Python scripts run on a local machine to produce cleaned/structured data.
3. Power BI Desktop connects to the cleaned dataset.
4. Final report is published to Power BI Service (for web or organizational access).

- **Component View**
    - Python ETL component.
    - Clean dataset (data model).
    - Power BI report file (.pbix).
    - Power BI Service (optional, for sharing with stakeholders).

---

6. Testing, Validation & Future Extensions

6.1 Testing & Validation

- Compare key numbers (e.g., total trips, average delay) between:
    - Python calculations.
    - Power BI measures.
- Spot-check random transactions to ensure:
    - Date/time and station mappings are correct.

    - Delay and refund flags are accurately represented.
- Validate KPIs with project supervisors or domain references when available.

6.2 Deployment Strategy

- Maintain a GitHub repository containing:
    - Python scripts (ETL & analysis).
    - Sample raw dataset (if allowed).
    - Cleaned dataset schema.
    - Documentation and screenshots of the Power BI dashboard.
- Publish dashboard to Power BI Service:
    - Decide whether access is public (if dataset allows) or restricted to certain users.

6.3 Conclusion & Future Work

This project demonstrates how data analytics can transform raw railway booking and journey records into meaningful operational insights.
By providing clear KPIs and interactive visualizations, decision makers can:

- Identify problematic routes and times.
- Monitor delay trends and their impact on refunds.
- Better understand customer purchasing and payment behaviors.

Possible future work includes:

- Adding predictive models (e.g., delay prediction using machine learning).
- Integrating real-time data feeds (APIs) for live dashboards.
- Expanding the dataset to include more years or additional regions.