Marie-Johanna Perli
Maarja Kovalevski
Hendrik Aruoja
Helen Kustavus

Project GitHub

University of Tartu
Data Engineering (LTAT.02.007)

05.10.2025

Understanding London City Bike Usage Through Weather Patterns

Table of Contents

Business Brief

Objective

To analyze London city bike usage patterns based on weather conditions, to understand how people use bikes during different weather conditions, whether weather impacts the journeys and how to use weather forecast to optimize operations. Ex: if we know rain is coming, to bring more bikes to the station. This analysis will be done on one bike station.

Stakeholders

- Transport for London (TfL) to optimize bike distribution and maintenance.
- Urban Planners and Policymakers to design bike-friendly infrastructure and policies.
- Operations & Logistics Teams to plan rebalancing schedules and fleet management.
- Data Science & Analytics Teams to monitor trends and develop predictive models.
- Environmental Agencies to assess the impact of weather-related transport trends.

Key Metrics (KPIs)

1. Daily/Hourly Bike Trip Volume – Total number of bike checkouts across stations.
2. Average Trip Duration – Time users spend on bikes, segmented by weather conditions.

Business Questions

1. How much does temperature affect the number of daily and hourly bike rentals?
2. How much precipitation and wind speed significantly reduce bike usage?
3. Can we forecast bike demand at different stations based on upcoming weather conditions?
4. Which times of day are most sensitive to weather changes in terms of bike usage?
5. What is the threshold of "bad weather" beyond which bike usage drops sharply?
6. How much does the weather change the average duration of rides?
7. How does weather affect ebike vs bike usage?

Datasets

1. London Bike Sharing dataset

- **Description**: The “London Bike-Share Usage” dataset on contains detailed records of bicycle journeys collected from Transport for London (TfL) for August 2023, including timestamps, start/end stations, trip durations, and rider types. It’s designed for analyzing bike-sharing patterns, forecasting demand, and building predictive models on urban mobility.
- **Source**: Kaggle - https://www.kaggle.com/datasets/kalacheva/london-bike-share-usage-dataset
- **Data**:
    - Event based London Bike-Share bike usage data
    - ~700,000+ daily records.
    - August 2023
- **Relevance**: Provides detailed trip-level data that enables analysis of demand, station usage, and ride patterns.

2. London (Heathrow) weather dataset

- **Description**: The UK hourly weather observation data contain meteorological values measured on an hourly time scale. The measurements of the concrete state, wind speed and direction, cloud type and amount, visibility, and temperature were recorded by observation stations operated by the Met Office [description from source]. Currently we used Heathrow datapoint as it was very consistent however the solution is scalable for weather stations all over London.
- **Source**: https://data.ceda.ac.uk/badc/ukmo-midas-open/data/uk-hourly-weather-obs/dataset-version-202507/greater-london/00708_heathrow
- **Data**:
    - Hourly weather data from various London observation points
    - 104 columns. i.e: Air temperature, Wind direction, Dew point temp
- **Relevance**: Offers granular weather conditions data that can be joined with bike ride timestamps and rain data to measure correlations between weather and demand.

3. London (Heathrow) rain dataset

- **Description**: The UK hourly rainfall data contain the rainfall amount (and duration from tilting syphon gauges) during the hour (or hours) ending at the specified time. [description from source]. Currently we used Heathrow datapoint as it was very consistent however the solution is scalable for weather stations all over London.
- **Source**: https://data.ceda.ac.uk/badc/ukmo-midas-open/data/uk-hourly-rain-obs/dataset-version-202507/greater-london/00708_heathrow

- **Weather data**
    - Hourly rain data from various London observation points
    - 15 columns. i.e: Observation time, Prescription amount
- **Relevance**: Offers granular rain conditions data that can be joined with bike ride timestamps and rest of the weather data to measure correlations between weather and demand.

Why These Datasets?

- All datasets satisfy the requirement: ≥ 1000 rows and ≥ 8 columns
- They are complementary: bike usage (demand-side) and weather (environmental factors)
    - Weather data was split - rain data is stored separately from the rest in the data source (likely due to the fact it is measured differently from the rest and not every weather station as the ability)
    - However, rainfall is an important metric in analysing bike usage in London, which is known for its rain amounts.
- Together, they allow exploration of key business questions

Tooling

Ingestion – Apache Airflow

Automation for data ingestion. Airflow schedules and manages data ingestion from CEDA. It extracts new data once a year, loads raw data into Snowflake, alerts of data errors.

Storage – Snowflake

Cloud based data warehouse for storing raw and edited data. Snowflake can scale easily and separates storage from computing, so data can be ingested, transformed, and analyzed at the same time without slowing things down.

Transformation – Data Build Tool

For transforming, cleaning, and modeling data inside Snowflake using SQL-based pipelines. Dbt models define transformation logic, SQL models make transformations transparent, dbt ensures data quality, dbt also provides automatic documentation.

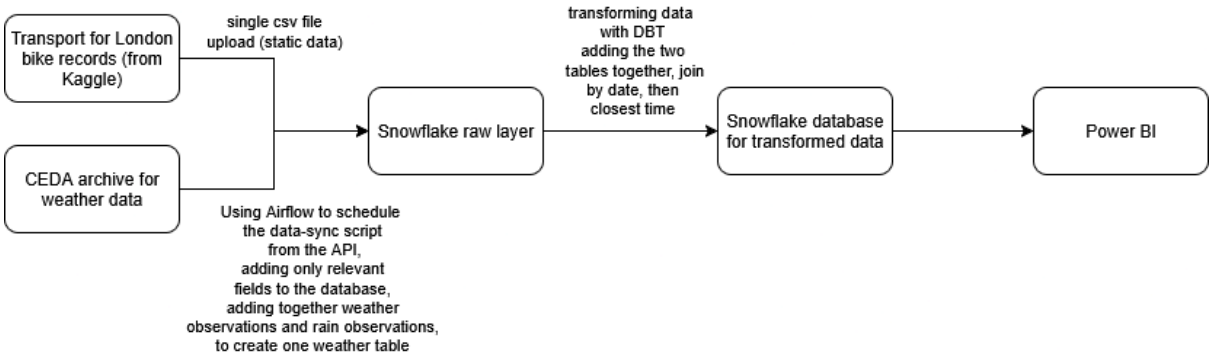
Serving – Snowflake (Data Serving Layer)

With Snowflake as the serving layer, you can explore datasets using Power BI, and you can set up role-based access so not everyone sees all the data (not essential for this project, but good to keep in mind). Snowflake delivers fast and scalable data access, helping ensure consistent reporting.

Analytics – Power BI

For visualisations and business reports. It connects directly to Snowflake, so if data is updated, it can show the newest data. We can create dashboards as needed, users can explore the data in a simple way.

Data Architecture



Update frequency: since the bike data is uploaded unpredictably, with random frequency and time, it seems reasonable to update the bike data and weather data once a year (bike data is not updated in Kaggle but in Transport for London website)

Data quality check:

- There must be precipitation every month (we had previous data where it didn't show rain at all, just 0 for the whole month, that was data error). In that case a human has to verify that this was correct for the month.
- Ride's number, timestamps, and weather observation ID and timestamp can't be NULL and must be unique
- No text in integer datatypes
- In bike data start date must be smaller (or equal) than end date, these values can't be NULL

Data Model

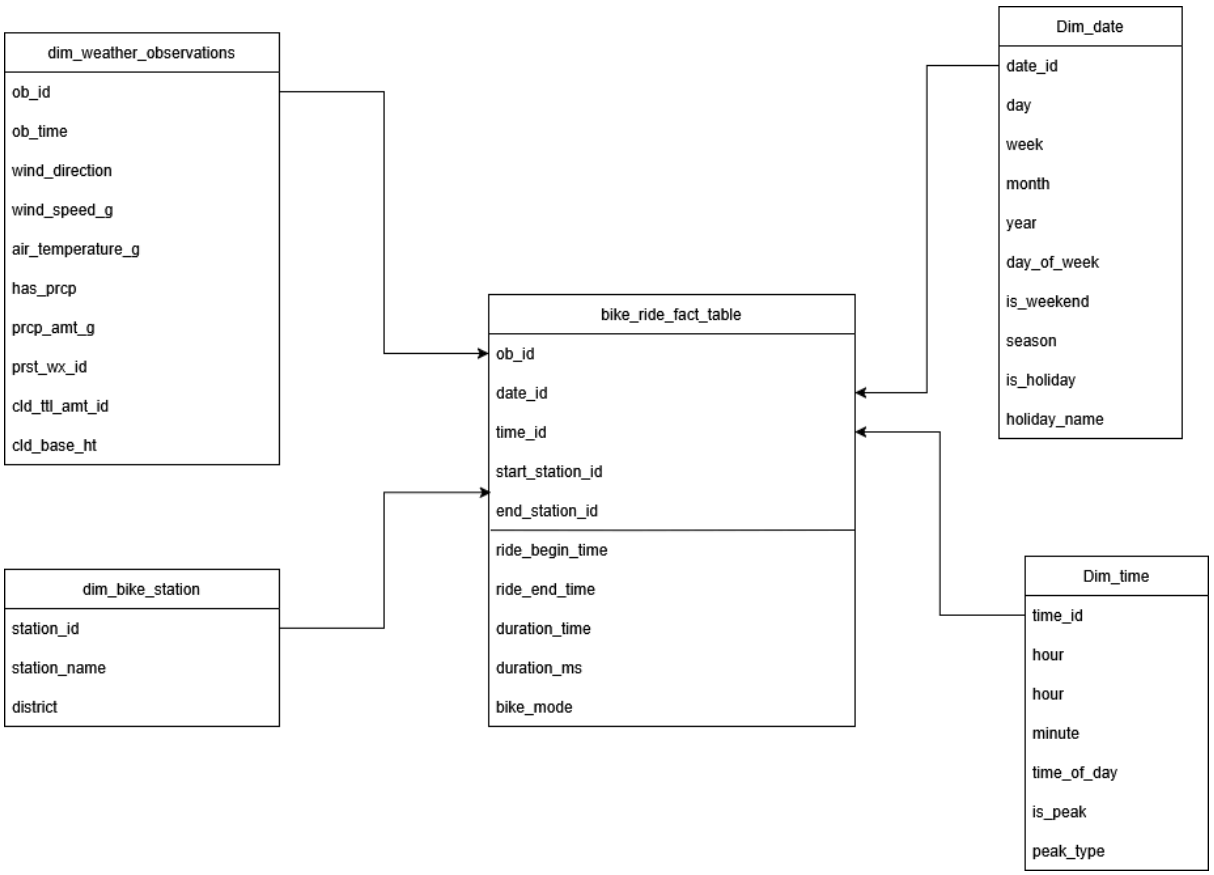
Slowly Changing Dimensions

Date data (Type 0) - Date attributes (like month, week, day, season) are fixed in time, so no history or Type 1/2 logic is needed.

Time data (Type 0) - Time attributes (like hour, minute, peak time, is weekend) are fixed in time, so no history or Type 1/2 logic is needed.

Bike station (Type 1) - bike station name history is not necessary in the scope of the project, since name change does affect the data value.

Weather data (Type 1) - since this is data that is collected from individual weather observations, the changes will only be implemented if data needs to be corrected.



Data dictionary

Bike data

Bikeshare data is also available through CityBikes API: https://publicapis.io/city-bikes-api. It is possible to change the data so we scrape the info from the API. This gives us station info and vehicle info separately. Since the vehicle info is currently unavailable for santander-cycles (London bikeshare service) then we decided to use the already available dataset with one-time scraping.

Bike rides - bike_ride_fact

The bike rides table is our fact table that stores information about the bike ride. Connection to weather, station, date and time information is made in the fact table.

Field name	Field	Datatype	Description
ob_id	Weather Observation ID	serial	UUID for the weather observation
date_id	Date ID	serial	UUID for the date
time_id	Time ID	serial	UUID for the time
start_station_id	Start station ID	serial	UUID for the start station
end_station_id	End station ID	serial	UUID for the end station
ride_begin_time	Start date	timestamp	Ride start timestamp
ride_end_time	End date	timestamp	Ride end timestamp
duration_time	Total duration	varchar(20)	Bikeride duration in minutes and seconds
duration_ms	Total duration (ms)	integer	Bikeride duration in milliseconds
bike_mode	Bike mode	varchar(20)	Bike mode (CLASSIC, PBSC_EBIKE, etc.)

Bike station data - dim_station

Base information about the bike station.

Field name	Field	Datatype	Description
station_id	Station ID	serial	Unique station ID
station_name	Station name	varchar(100)	Station name
district	District	varchar(100)	(OPTIONAL IF WE GET THE DATA) District where station is located

Time data - dim_time

Base information about bikes.

Field name	Field	Datatype	Description
time_id	Time ID	serial	Unique time identifier
hour	Hour	integer (2)	Hour of the day (0–23)
minute	Minute	integer (2)	Minute of the hour (0–59)
time_of_day	Time of day	enum	Part of the day. Values: MORNING AFTERNOON EVENING NIGHT
is_peak	Is peak hour	boolean	True if time falls within peak traffic hours. Peak hours are 06:30 to 09:30 and 16:00 to 19:00 (according to tfl)
peak_type	Peak hour type	enum (non mandatory)	Category of peak time. Values: MORNING EVENING

Date data - dim_date

Base information about bikes.

Field name	Field	Datatype	Description
date_id	Date ID	serial	Unique time identifier
day	Day	integer (2)	Day of the month (1–31)
week	Week	integer (2)	Week number of the year
month	Month	integer (2)	Month number (1–12)
year	Year	integer (4)	Calendar year
day_of_week	Day of Week	enum	Name of weekday. Values: MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY
is_weekend	Is weekend	boolean	True if Saturday/Sunday, else False.
season	Season	enum	Season of the year. Based on months (winter - december to february, spring - march to may, summer - june to august, autumn - september to november). Values: SPRING SUMMER AUTUMN WINTER
is_holiday	Is holiday	boolean	True if the date is an official holiday False. 2023 holidays: 01.01.2023 07.04.2023

			10.04.2023 01.05.2023 29.05.2023 28.08.2023 25.12.2023 26.12.2023
holiday_name	Holiday name	enum (non mandatory)	Name of the holiday if applicable, else NULL. Values: 01.01.2023 - New Year's Day 07.04.2023 - Good Friday 10.04.2023 - Easter Monday 01.05.2023 - Early May Bank Holiday 29.05.2023 - Spring Bank Holiday 28.08.2023 - Summer Bank Holiday 25.12.2023 - Christmas Day 26.12.2023 - Boxing Day

Weather data - dim_weather_observations

Weather data is also available through OPeNDAP API: https://archive.ceda.ac.uk/tools/. It is possible to change the data so we scrape the info from the API. Currently we download using links provided above for creating the base.

Weather observation - weather_observation table

The weather observations table stores historical timestamped weather readings. Each reading has its own UUID assigned. The weather data table has two sources: weather observation and rain observation. Both are hourly based from CED and joined based on hourly timestamp.

The actual raw data consists of 104+15 fields, most of which we are not using. We will transform it into a given format and we decided not to include all fields into this dictionary as we plan to do the data transformation before storing and the current table is for stored data.

Field	**Long name**	**Datatype**	**Description**
ob_id	Observation ID	serial	UUID for the observation. Given by us.

ob_time	Observation timestamp	timestamp	Observation date and time. Format: YYYY-MM-DD HH:MM:SS
wind_direction	Wind direction groups	enum	Wind direction (data transformation needed). Values: N 337.5° – 360° OR 0° – 22.5° NE 22.5° – 67.5° E 67.5° – 112.5° SE 112.5° – 157.5° S 157.5° – 202.5° SW 202.5° – 247.5° W 247.5° – 292.5° NW 292.5° – 337.5°
wind_speed	Wind speed	enum	Wind speed description (data transformation needed). Values: < 2 - CALM 2–5 - LIGHT WIND 6–10 - MODERATE > 10 - STRONG Use wind_speed_unit_id + wind_speed values from data to generate values
air_temperature_g	Air temperature groups	enum	Air temperature descriptions (data transformation needed). Values: < 0 - FREEZING 0–10 - COLD 10–20 - MILD 20–25 - COMFORTABLE 25–30 - WARM > 30 - HOT Use air_temperature in data to create the values
has_prcp	Boolean	boolean	Shows if it was raining.

			If prcp_amt >0 -> TRUE, else FALSE
prcp_amt_g	Precipitation amount	enum	Precipitation amount (data transformation needed). Values: 0 - NO RAIN 0–2 - LIGHT RAIN 2–10 - MODERATE RAIN > 10 - HEAVY RAIN Use prcp_amt in data to create the values
prst_wx_id	Weather description from the original observation. WMO code	int (2)	Mapping table 4677 https://artefacts.ceda.ac.uk/badc_datadocs/surface/code.html
cld_ttl_amt_id	Total clouds amount based on WMO code	int (2)	Mapping table 2700 https://artefacts.ceda.ac.uk/badc_datadocs/surface/code.html
cld_base_ht	Minimum clouds height	int(5)	Sellele peaks ka mingid vahemikud tegema

**prcp_amt is the only field which source is the uk-hourly-weather-rain-obs dataset instead of the uk-hourly-weather-obs dataset. Both sets are hourly based, in the same format and from the same data provider and cover weather data so it made sense to store them in the same table. There should be no effect on rest of the data as well in case of availability issues with one dataset.*

Demo Queries

How much does temperature affect the number of daily and hourly bike rentals?
Hourly demo query (works the same if we join by date instead of hour):

```
SELECT COUNT(br.ride_id) AS total_rides, wo.air_temperature
FROM weather_observations_bike_ride_station AS wobrs
LEFT JOIN weather_observations AS wo
        ON wo.ob_id = wobrs.ob_id
LEFT JOIN bike_ride AS br
        ON br.ride_id = wobrs.ride_id
        AND DATE_TRUNC('hour', br.begin_time) = DATE_TRUNC('hour, wo.ob_time)
GROUP BY wo.air_temperature
ORDER BY total_rides DESC;
```

How much precipitation and wind speed significantly reduce bike usage?
Hourly info demo query:

```
SELECT COUNT(br.ride_id) AS total_rides, wo.wind_speed, wo.prcp_amt
FROM weather_observations_bike_ride_station AS wobrs
LEFT JOIN weather_observations AS wo
        ON wo.ob_id = wobrs.ob_id
LEFT JOIN bike_ride AS br
        ON br.ride_id = wobrs.ride_id
        AND DATE_TRUNC('hour', br.begin_time) = DATE_TRUNC('hour, wo.ob_time)
GROUP BY wo.wind_speed, wo.prcp_amt
ORDER BY total_rides DESC;
```

Can we forecast bike demand at different stations based on upcoming weather conditions?

```
SELECT COUNT(br.ride_id) AS total_rides, wobrs.station_id, br.station_name, wo.wind_speed, wo.wind_direction, wo.prcp_amt, wo.msl_pressure, wo.dewpoint, wo.drv_hr_sun_dur
FROM weather_observations_bike_ride_station AS wobrs
LEFT JOIN weather_observations AS wo
        ON wo.ob_id = wobrs.ob_id
LEFT JOIN bike_ride AS br
        ON br.ride_id = wobrs.ride_id
        AND DATE_TRUNC('hour', br.begin_time) = DATE_TRUNC('hour, wo.ob_time)
LEFT JOIN bike_station AS bs
        ON bs.station_id = wobrs.station_id
GROUP BY wobrs.station_id, wo.wind_speed, wo.wind_direction, wo.prcp_amt, wo.msl_pressure, wo.dewpoint, wo.drv_hr_sun_dur
ORDER BY wobrs.station_id;
```

Which times of day are most sensitive to weather changes in terms of bike usage?

```
SELECT COUNT(br.ride_id) AS total_rides, wo.wind_speed, wo.wind_direction, wo.prcp_amt, wo.msl_pressure, wo.dewpoint, wo.drv_hr_sun_dur
FROM weather_observations_bike_ride_station AS wobrs
LEFT JOIN weather_observations AS wo
```

```
        ON wo.ob_id = wobrs.ob_id
LEFT JOIN bike_ride AS br
        ON br.ride_id = wobrs.ride_id
        AND DATE_TRUNC('hour', br.begin_time) = DATE_TRUNC('hour, wo.ob_time)
GROUP BY wo.wind_speed, wo.wind_direction, wo.prcp_amt, wo.msl_pressure,
wo.dewpoint, wo.drv_hr_sun_dur;
```

What is the threshold of "bad weather" beyond which bike usage drops sharply?

```
SELECT COUNT(br.ride_id) AS total_rides, wo.wind_speed, wo.prcp_amt,
wo.drv_hr_sun_dur
FROM weather_observations_bike_ride_station AS wobrs
LEFT JOIN weather_observations AS wo
        ON wo.ob_id = wobrs.ob_id
LEFT JOIN bike_ride AS br
        ON br.ride_id = wobrs.ride_id
        AND DATE_TRUNC('hour', br.begin_time) = DATE_TRUNC('hour, wo.ob_time)
GROUP BY wo.wind_speed, wo.prcp_amt, wo.drv_hr_sun_dur
ORDER BY wo.wind_speed ASC, wo.prcp_amt ASC, wo.drv_hr_sun_dur DESC;
```

How much does the weather change the average duration of rides?

```
SELECT COUNT(br.ride_id) AS total_rides, br.duration_time, wo.wind_speed, wo.prcp_amt,
wo.drv_hr_sun_dur
FROM weather_observations_bike_ride_station AS wobrs
LEFT JOIN weather_observations AS wo
        ON wo.ob_id = wobrs.ob_id
LEFT JOIN bike_ride AS br
        ON br.ride_id = wobrs.ride_id
        AND DATE_TRUNC('hour', br.begin_time) = DATE_TRUNC('hour, wo.ob_time)
GROUP BY wo.wind_speed, wo.prcp_amt, wo.drv_hr_sun_dur
ORDER BY br.duration_ms DESC;
```

How does weather affect ebike vs bike usage?

```
SELECT COUNT(br.ride_id) AS total_rides, b.bike_mode, wo.wind_speed, wo.prcp_amt,
wo.drv_hr_sun_dur
FROM weather_observations_bike_ride_station AS wobrs
LEFT JOIN weather_observations AS wo
        ON wo.ob_id = wobrs.ob_id
LEFT JOIN bike_ride AS br
        ON br.ride_id = wobrs.ride_id
        AND DATE_TRUNC('hour', br.begin_time) = DATE_TRUNC('hour, wo.ob_time)
LEFT JOIN bike AS b
        ON b.bike_id = wobrs.
GROUP BY b.bike_mode, wo.wind_speed, wo.prcp_amt, wo.drv_hr_sun_dur
ORDER BY total_rides DESC;
```

Roles and efforts

Below are the main task distributions between members (and percentage of the total of given task):

Helen Kustavus - project management (95%), weather data (70%), data dictionary (60%) and star schema (95%). Note - did not work on the business brief part nor with the bike data.

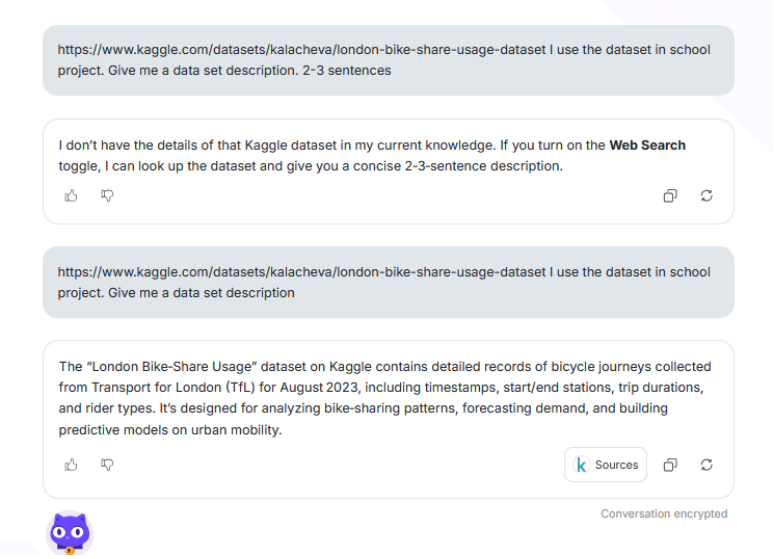
Maarja Kovalevski - data dictionary (40%), demo queries (100%), technical management (75%)

Hendrik Aruoja - datasets (35%) and tooling (40%)

Marie-Johanna Perli - datasets (15%), tooling (60%), data architecture (95%)

For every task or task part (leftover percentage) that is not covered above, we consider that all remaining members of the team contributed equally (divide leftover percentage with leftover members).

LLM usage

Member	LLM	Link
Helen	Proton Lumo (Lumo does not provide link sharing)	
Hendrik	ChatGPT	https://chatgpt.com/share/68d787a3-fc98-8000-be10-4e003826bdd0
Marie	ChatGPT 5	https://chatgpt.com/share/68de9229-a8f0-8009-b7c7-1acd6e940afc

Helen	ChatGPT	https://chatgpt.com/c/69027052-b37c-832e-b159-5e9d0a798b6d