**DIMTS Time Dashboard**

1. **Static Feeds Specification**

The following static feeds were used for fetching and displaying relevant information onto the dashboard

1. **Stops:-** List of all active stops and terminals

| **Field Name** | **Description** |
| --- | --- |
| stop\_id | Contains the unique identifier for each stop |
| stop\_code | Text that identifies the location for stop |
| stop\_name | Name of the location |
| stop\_lat | Latitude of the location |
| stop\_lon | Longitude of the location |
| zone\_id | Used for identification for the fare zone of the stop |

1. **Routes:-** Information about all the routes plying in Delhi

| **Field Name** | **Description** |
| --- | --- |
| route\_id | Contains the unique identifier for each route |
| agency\_id | Identifier for agency for the route |
| route\_short\_name | Short name of the route, doesn’t give any indication to places the route serves |
| route\_long\_name | Full name of the route |
| route\_type | Indicates what transportation can be used on the route |

1. **Trips:-** Frequency and schedules of buses on the routes

| **Field Name** | **Description** |
| --- | --- |
| route\_id | Contains the unique identifier for each route |
| service\_id | Identifies the set of dates when the service is available |
| trip\_id | Unique identifier for a trip on a route |
| shape\_id | Id that describes the vehicle travel path |

1. **Stop times:-** Arrival time of buses on a stop for a trip

| **Field Name** | **Description** |
| --- | --- |
| trip\_id | Unique identifier for a trip on a route |
| arrival\_time | Arrival time at a specific stop for a trip on a route |
| deparature\_time | Departure time from a specific stop for a trip on a route |
| stop\_id | Identifies the serviced stop |

**2. Live feed Details**

The Live GPS data for the vehicles plying on the route was received from the endpoint.

The data is being received in JSON format and is a list of dictionaries.

Following is the information that can be received from the endpoint.

| **Field Name** | **Description** |
| --- | --- |
| vehicle\_id | The registration number of a bus |
| agency | Name of the agency that operates on the routes |
| route\_long\_name | Full name of the route |
| route\_id | Contains the unique identifier for each route |
| upcoming\_stop\_idx | The sequentially generated index for a stop on the route |
| upcoming\_stop\_id | Stop\_id pertaining to a stop on a route |
| eta | The live estimated time for the arrival of a bus on a route |
| stops\_in\_btw | No. of stops between the live location of the bus and that particular stop |

**3. Database specification**

A database “project.db” was created to store the static data that was needed for the dashboard. The database created is a RDBMS implemented using python library SQLITE3.

Following are the tables:-

**i) STOPS** : (stop\_id, stop\_name) for mapping stop names to a stop id.

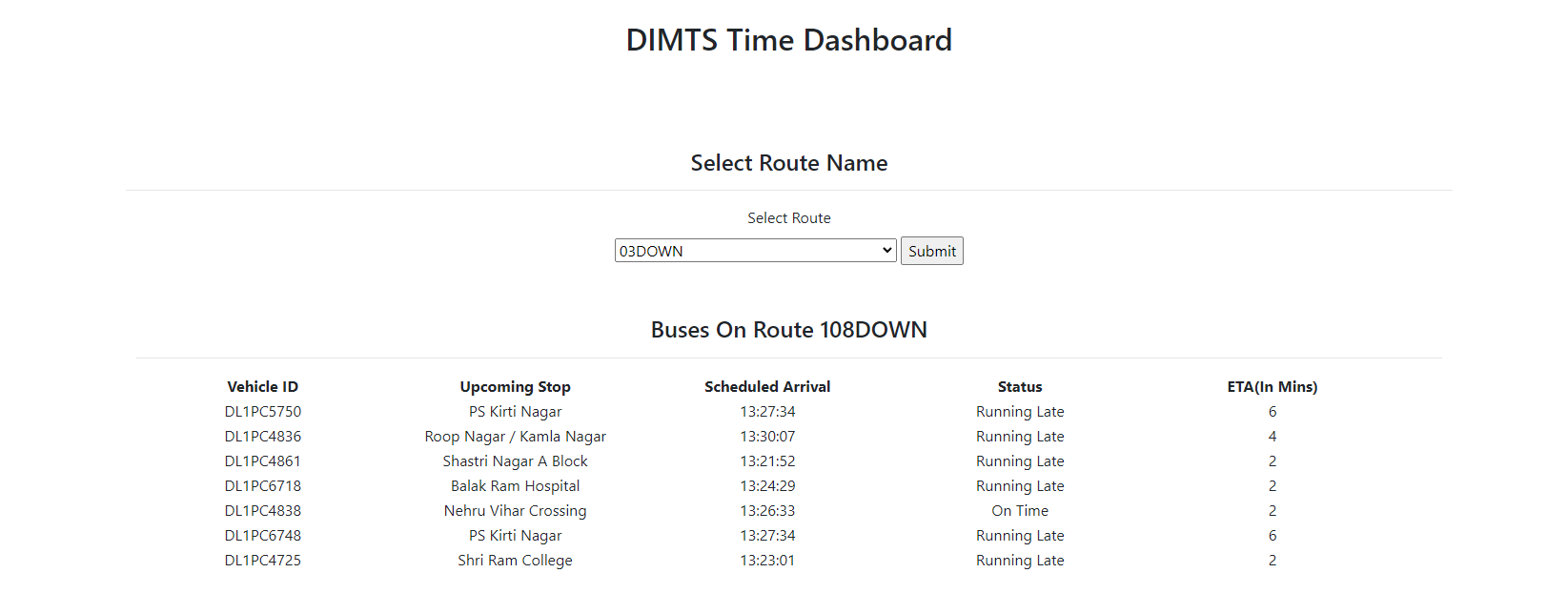
**ii) STOP\_TIME**: (trip\_id, arrival\_time, stop\_id, route\_id) for obtaining the arrival time for a particular stop for a trip on a route.

**iii) TRIPS**: (route\_id, trip\_id) for mapping the routes to trip.

**4. Implementation**

Through the dashboard we are presenting the user to see the current status of the buses on a particular route, their upcoming stop and the ETA.

Beside this the users can see the number of trips that are ongoing and the trips which are missed.

****

**->** The live data is fetched from the endpoint and is converted to a dataframe and only the required columns are being extracted from the live feed.

-> From the live data frame we are fetching the list of unique route names on which the buses are currently plying. These routes are then presented to the user as an option to be selected from a dropdown list.

-> Once a user selects a route of his/her choice, a list of the vehicles and other relevant information is extracted.

-> Now for each vehicle on the route the status of the trip is calculated and appended to a dictionary.

-> This final data in the form of a dictionary is sent to the html webpage for rendering and the details are presented on the frontend.

**5. Appendix**

* **Library Used:-**flask, pandas, requests, json, datetime, sqlite3
* **Endpoint**:-<http://192.168.26.172:14002/all_rt_buses_data>
* **StaticData:-** [**Open Transit Data | Delhi**](https://otd.delhi.gov.in/data/static/)