Completion Report
IoT Driven Smart Bus Stops

Project Guides

Dr P. Mirunalini, Assoc. Professor, CSE
Dr B. Bharathi, Assoc. Professor, CSE
Dr Cherry M P, Asst. Professor, English

Project Students

Karthik Desingu, B.E. CSE
Daniel Mark Isaac, B.E. ECE

Presentation Overview

Recent progress

- Literature Survey for prediction model
- Data Preprocessing for model
- IoT Time-Logger

Difficulties

- Difficulty in acquiring hardware for IoT
- Delayed delivery of components
- Difficulty in collaborating with team-mate
- Lack of a well-structured dataset of bus-arrival times

The Motivation behind our **Idea**

Indefinite Waiting

Commuters are kept in the dark.

Move back-and-forth to read bus numbers

Impractical Signage

Signage that can't be read from far.

Only in the local language

NOT Specially-Abled-Friendly

No GTFS for Buses in India!

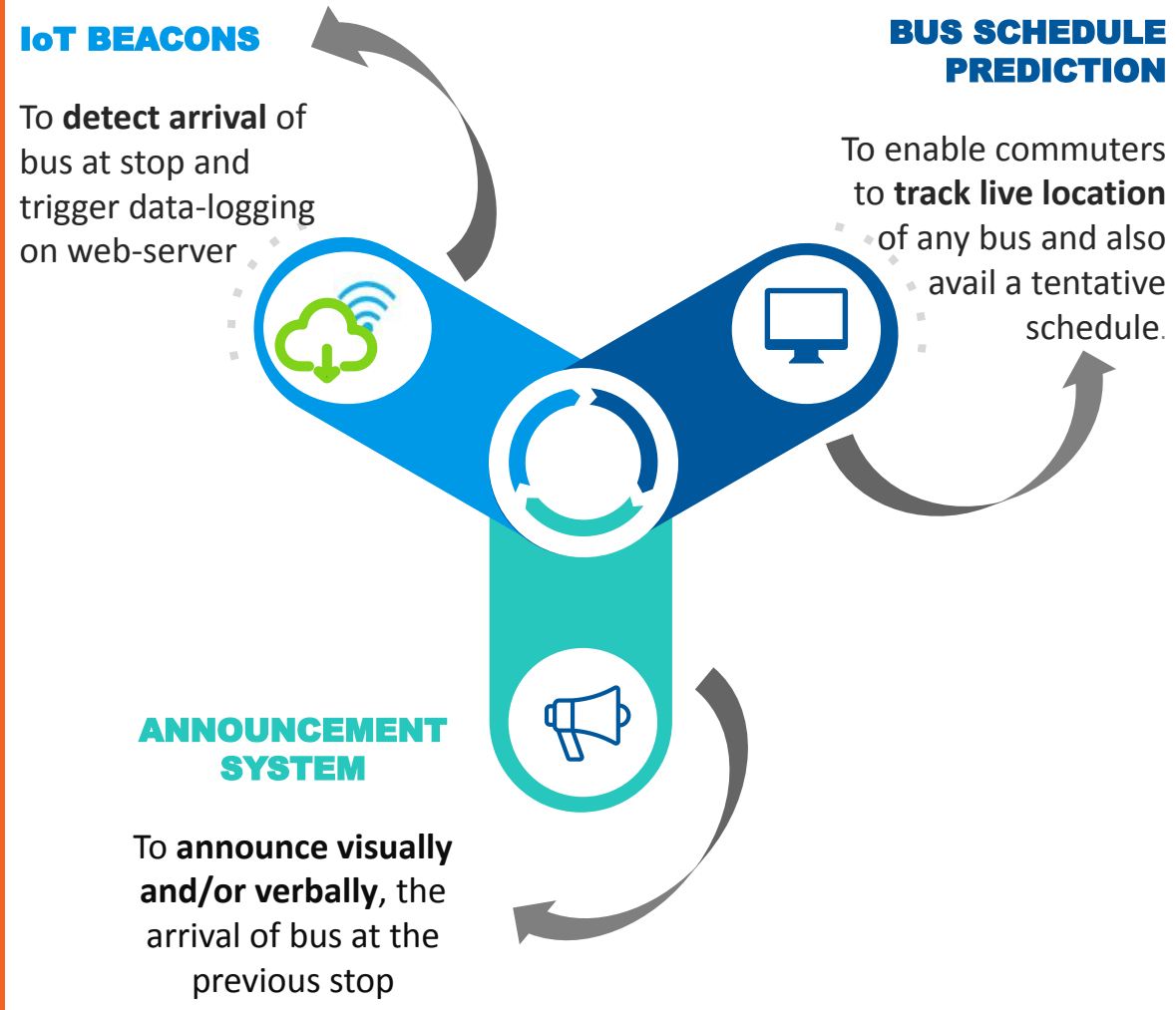
Bus schedules are not easily available to the public

Inaccurate Schedules

Seldom follow the prescribed schedule. Delays are natural. Need for prediction

Project Overview

- An **Internet of Things** and **Machine Learning** based system to **track, announce and predict** schedule of buses.
- **Live tracking** of buses on a dedicated website.
- Using collected data to build a **predictive model** to provide a tentative schedule.



Key Milestones

- **Data for Prediction Model Collected**
  Made frequent API-Queries to a Realtime-GTFS & logged responses
- **Data Preprocessing for Prediction Model**
  Cleaned the logged responses and extracted features
- **Literature Survey & Bare-Bones of Prediction Model**
  Surveyed existing methods, adopted ideas and laid down the bare-bones for the arrival-time prediction model
- **Database Designed**
  Remote-DB for data about buses, routes, stops, time-logs, etc

Key Milestones

→ **IoT Simulation Prepared**

As a proof of concept, using the Proteus software

→ **Time-Logger IoT Built**

To log arrival times of buses at each stop

→ **Website Developed**

To facilitate tracking and planning commutes online

→ **API Server Set-Up**

To establish a communication channel for the bus-transit system

Key Milestones

➔ Implementation of ML Model

The presented idea will be concretized through data visualization, trained, tested and improved

➔ IoT for Display System

LCD display in every stop to announce the next buses arriving at stop along with expected time

➔ Integration of IoT, Remote-DB and ML-Model

Ultimately, the Hardware; Web-server and Database; and the Prediction Model will work in conjunction

Data Collection

Made API-Queries to a Realtime-GTFS and Logged Responses

Massachusetts Bay Transportation Authority

GTFS

- **MBTA** offers a **Realtime-GTFS** over HTTP and **Protocol-Buffer** based API endpoints
- Regular queries made to collect bus **live running status** and **alerts**
- Responses collected from **Oct 2020** to **Jan 2021,** for several routes at each bus stop

GTFS - What and Why

- A **standard format** for publishing public transit data
  - Dedicated communities and wikis to collect GTFS providers
- Two variants: **Realtime** and **Static**
  - Realtime**:** APIs for **live** GPS-based **updates and alerts**
  - Static**: Highly-structured** CSV Repository of **transit-schedules**
- Hardly any datasets exist for bus arrival time logs
  - Even existing ones are **irregular** & **unorganized**. Not usable
- More recently, transit authorities have **started adopting GTFS** for public transparency and ease-of-access

```
header {
  gtfs_realtime_version: "2.0"
  autority: "MBTA GTFS-RT"
}

entity {
  id: "816771"
  trip_update {
    trip {
      trip_id: "50031875"
    }
    stop_time_update {
      stop_sequence: 3
      arrival {
        delay: 389
      }
    }

    stop_time_update {
      stop_sequence: 8
      arrival {
        delay: 166
      }
    }

    stop_time_update {
      stop_sequence: 10
    }
  }
}
```

Snapshot of Raw-Data

MBTA's Realtime-GTFS

- **Protocol Buffer API** response for **Trip-Updates**
- **Collected** and **stored** for every API request
- Similar response for **service alerts** - cancellation of service, truncated service, etc
-

Collected from **Oct 2020** to **Jan 2021**

Data Preprocessing

Cleaning and extraction of features from collected data

Massachusetts Bay Transportation Authority

GTFS

- **MBTA Static-GTFS** is used to understand their transit-network
- Corrupt values are discarded
- **Responses** are processed into **CSV file(**s)

Snapshot of Processed-Data

ServiceDate	Route	Direction	HalfTripId	Stop	Timepoint	TimepointOrder	PointType	StandardType	Scheduled	Actual
2020-10-01 00:00:00.000	SL1	Inbound	48893498	17091	terma	1	Startpoint	Schedule	1900-01-01 05:38:00.000	1900-01-01 05:45:38.000
2020-10-01 00:00:00.000	SL1	Inbound	48893498	17093	trmb2	3	Midpoint	Schedule	1900-01-01 05:40:00.000	1900-01-01 05:47:05.000
2020-10-01 00:00:00.000	SL1	Inbound	48893498	17095	terme	5	Midpoint	Schedule	1900-01-01 05:44:00.000	1900-01-01 05:49:40.000
2020-10-01 00:00:00.000	SL1	Inbound	48893498	12007	twtnn	6	Midpoint	Schedule	1900-01-01 05:46:00.000	1900-01-01 05:51:41.000
2020-10-01 00:00:00.000	SL1	Inbound	48893498	12008	twtns	7	Midpoint	Schedule	1900-01-01 05:49:00.000	1900-01-01 05:54:02.000
2020-10-01 00:00:00.000	SL1	Inbound	48893441	17091	terma	1	Startpoint	Headway	1900-01-01 05:53:00.000	1900-01-01 05:56:40.000
2020-10-01 00:00:00.000	SL1	Inbound	48893441	27092	trmb1	2	Midpoint	Headway	1900-01-01 05:54:00.000	1900-01-01 05:58:01.000
2020-10-01 00:00:00.000	SL1	Inbound	48893441	17093	trmb2	3	Midpoint	Headway	1900-01-01 05:55:00.000	1900-01-01 05:59:15.000
2020-10-01 00:00:00.000	SL1	Inbound	48893498	74614	conrd	8	Midpoint	Schedule	1900-01-01 05:56:00.000	1900-01-01 05:57:40.000
2020-10-01 00:00:00.000	SL1	Inbound	48893441	17094	trmcd	4	Midpoint	Headway	1900-01-01 05:57:00.000	1900-01-01 06:01:18.000
2020-10-01 00:00:00.000	SL1	Inbound	48893441	17095	terme	5	Midpoint	Headway	1900-01-01 05:59:00.000	1900-01-01 06:03:29.000
2020-10-01 0						11	Endpoint	Schedule	1900-01-01 06:01:00.000	1900-01-01 06:02:17.000
2020-10-01 0						6	Midpoint	Headway	1900-01-01 06:01:00.000	1900-01-01 06:05:12.000

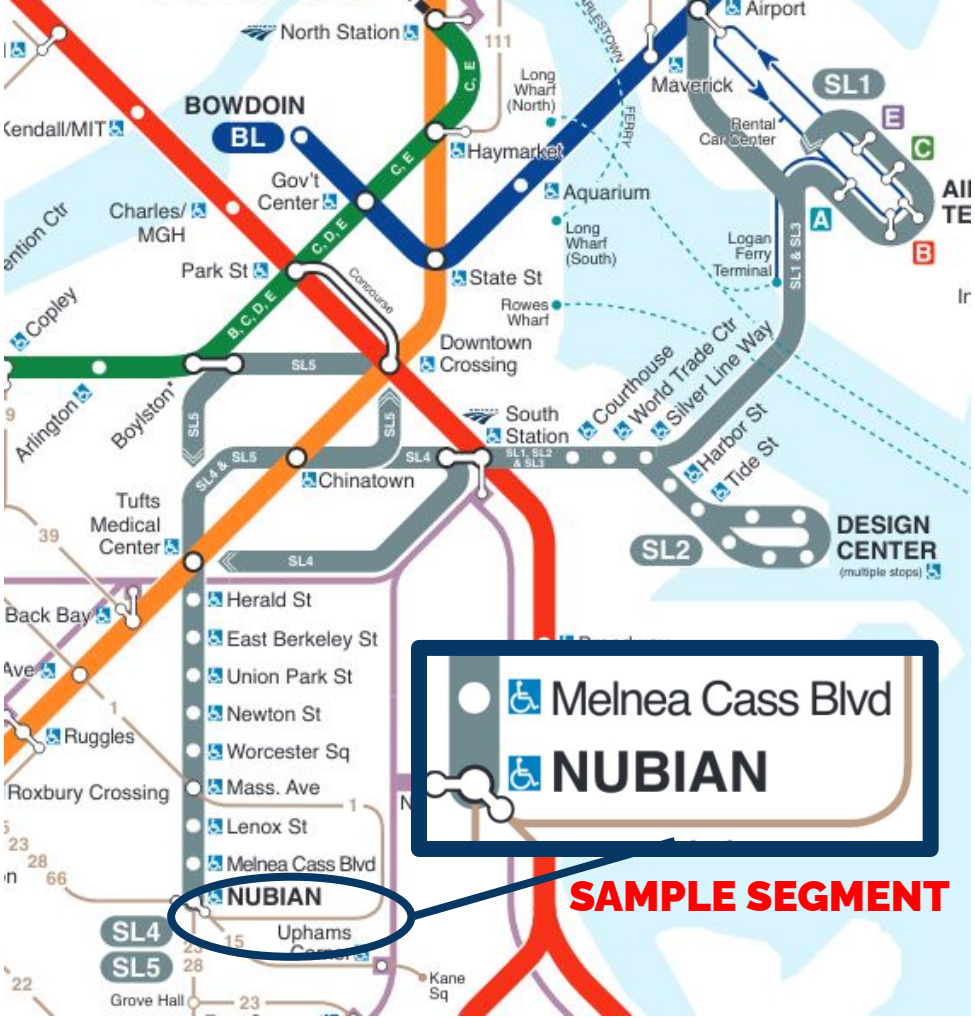
- **Protocol Buffer** based API responses collected and stored in **CSV format**
- Selected routes and stops
- Raw-data for model training

About 400MB of **CSV** data available before cleaning for

Bare-Bones of Prediction Model

Based on a **literature survey** of existing prediction methods

- **Literature Survey** was done to study existing **prediction models** for bus arrival predictions
- **Segment-wise** travel-time prediction model will be developed
- Historical Data + Present Situation = **Dynamic Model**
- **MBTA Bus Network** will be used to train and test the model



- The **Silver Lines:** SL1, SL2, SL4 and SL5 are taken up
- **Segment:** The stretch of road between two consecutive stops
- Develop model to predict the travel-time in a particular segment
- **Treat all routes** running in the **segment** as

A **Hybrid Model** is Ideated

HISTORIC DATA

Time of Day
Incremental Trend Curve

Recent Travel-Times
Of multi-routes on segment
Exponential Weighted Avg.

Holiday / Working Day
If significant distinction exists, create two diff. trend curves

REAL TIME OBSERVATIONS

Deviation b/w predicted & actual times
On the given day (or) interval of time

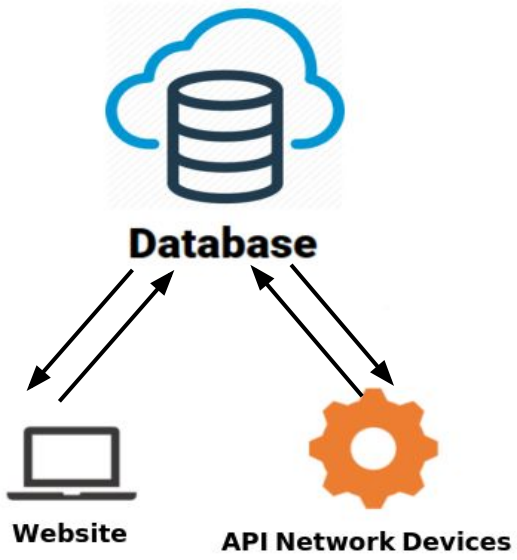
Queuing of Buses at Stop
Increases passenger pickup time

- Incorporate present-day scenario with the general trend
- Improve accuracy of prediction
- Sensitive to sudden changes

Baseline Prediction + **Numeric Realtime Factor** = **DYNAMIC PREDICTION**

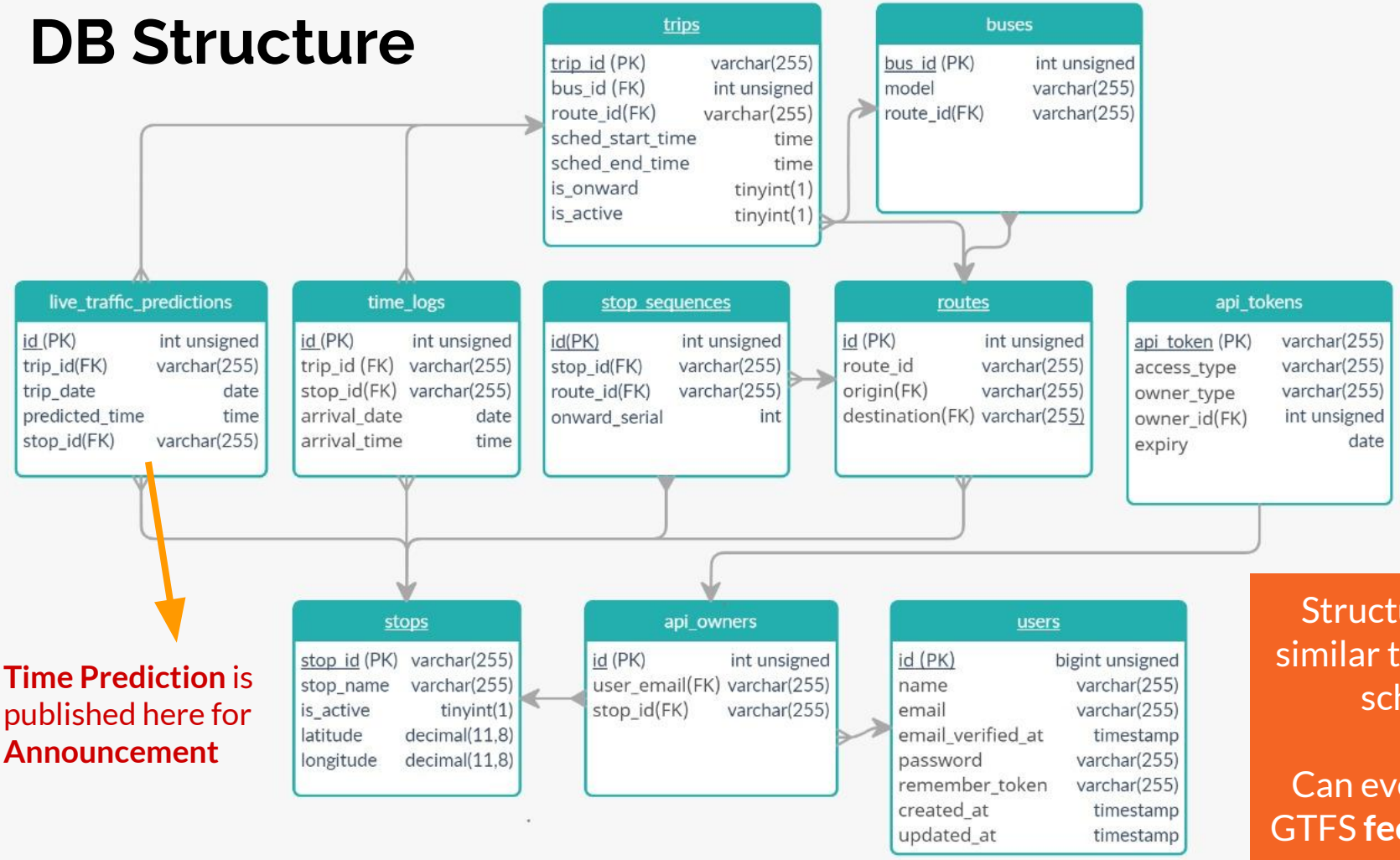
Remote Database

Data repository of buses, routes, stops, time-logs, etc



- **Heart** of the **Publish-Subscribe** architecture
- Accessible through **: Website & API**
- **Synchronizes** data and operations over the transit-network
- Can evolve into a **GTFS** data collection and feed network repository

DB Structure



Time Prediction is published here for **Announcement**

Structured very similar to the **GTFS** scheme.

Can evolve into a GTFS **feed network**

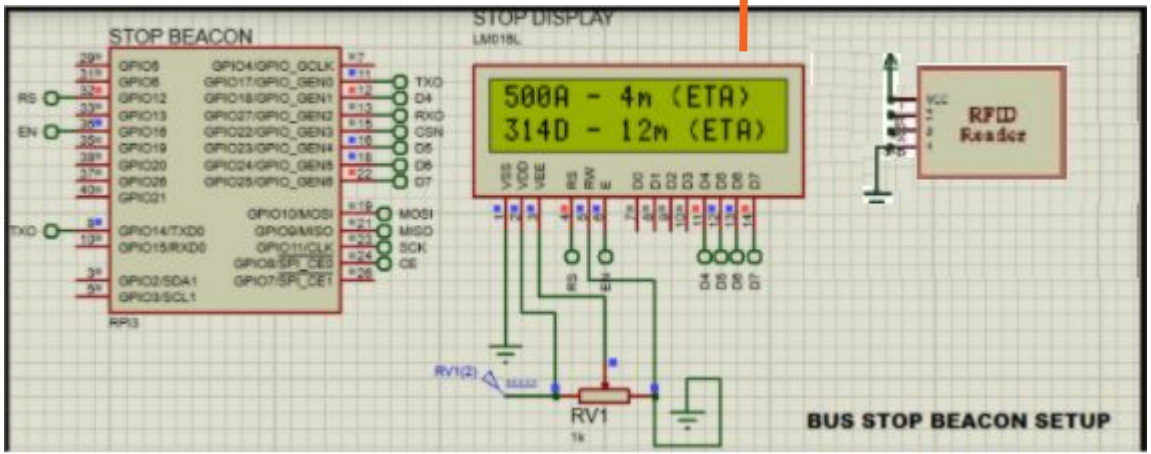
IoT Simulation

Software simulation for proof of concept

- The IoT concept is simulated using Proteus
- Difficulty in acquiring parts during the Pandemic
- To test the success of the concept

SIMULATION USING RFID::

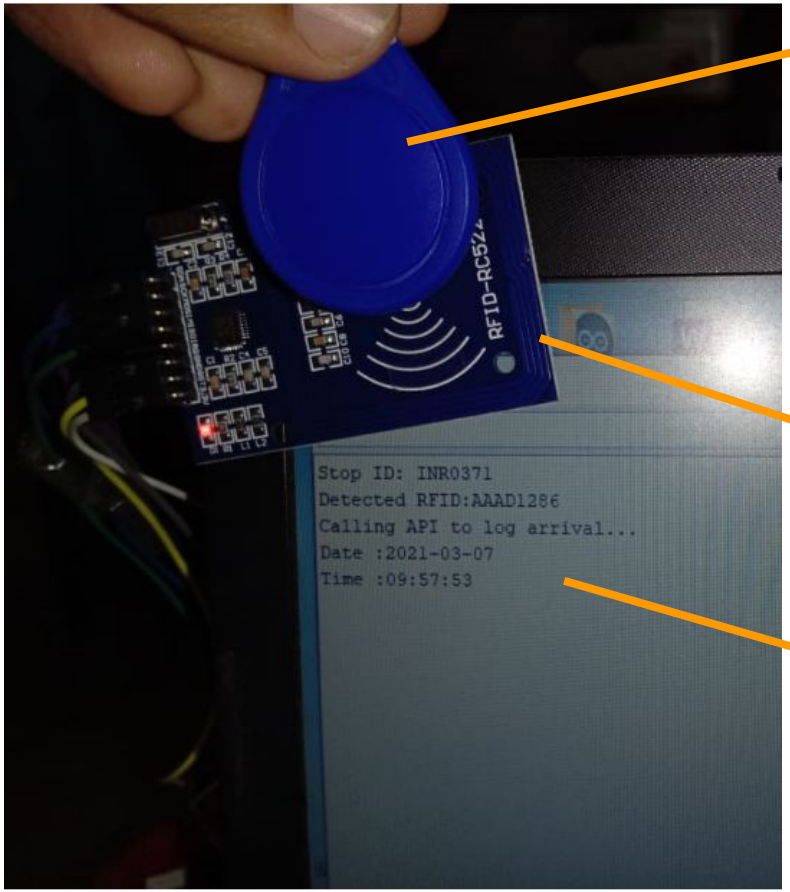
Reception of trip code and ETA details



Time Logger

- Hardware Implementation of the software simulation
- To Log bus id and time details

Time getting logged by RFID scan



RFID

RFID Scanner

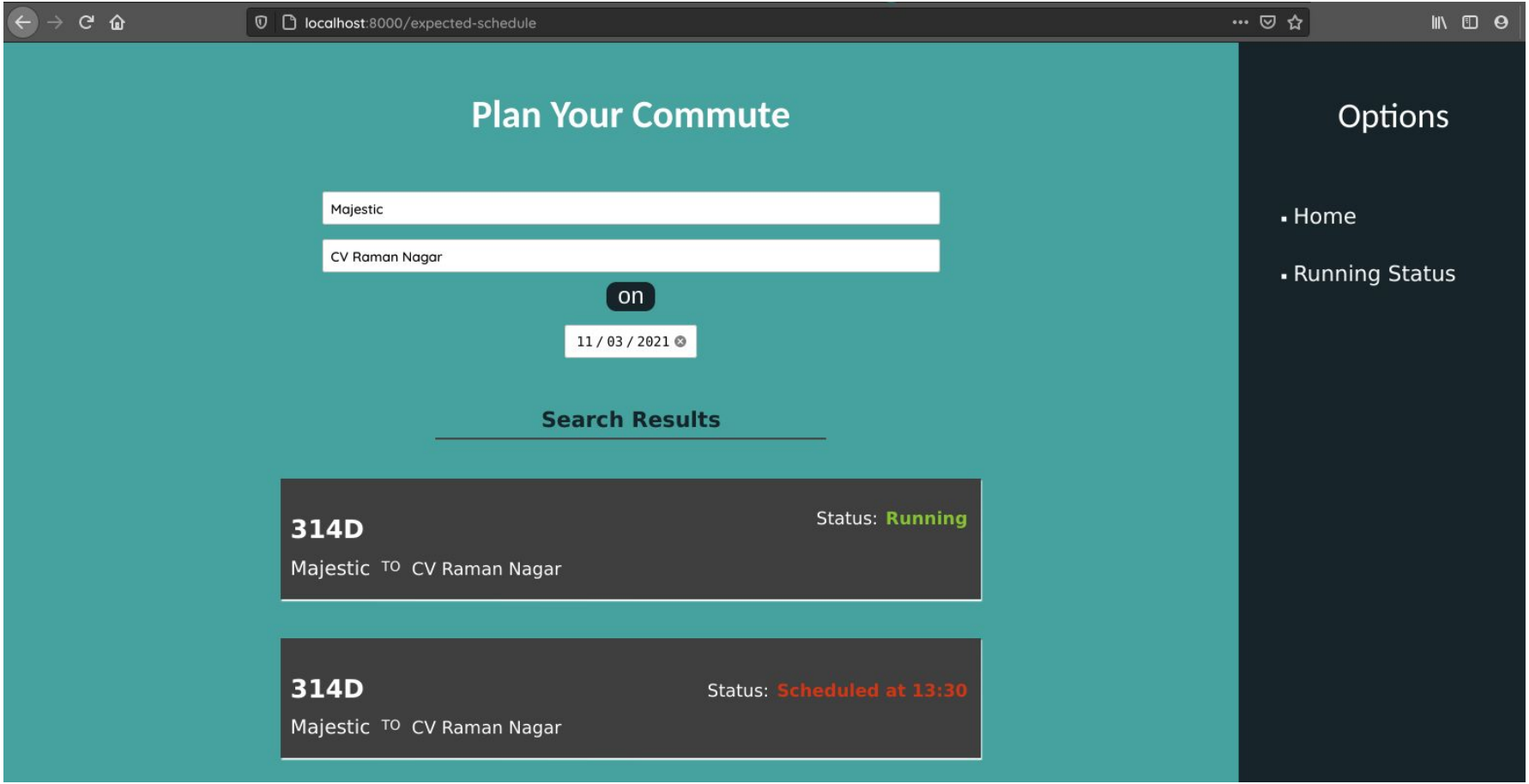
Time getting logged

Website

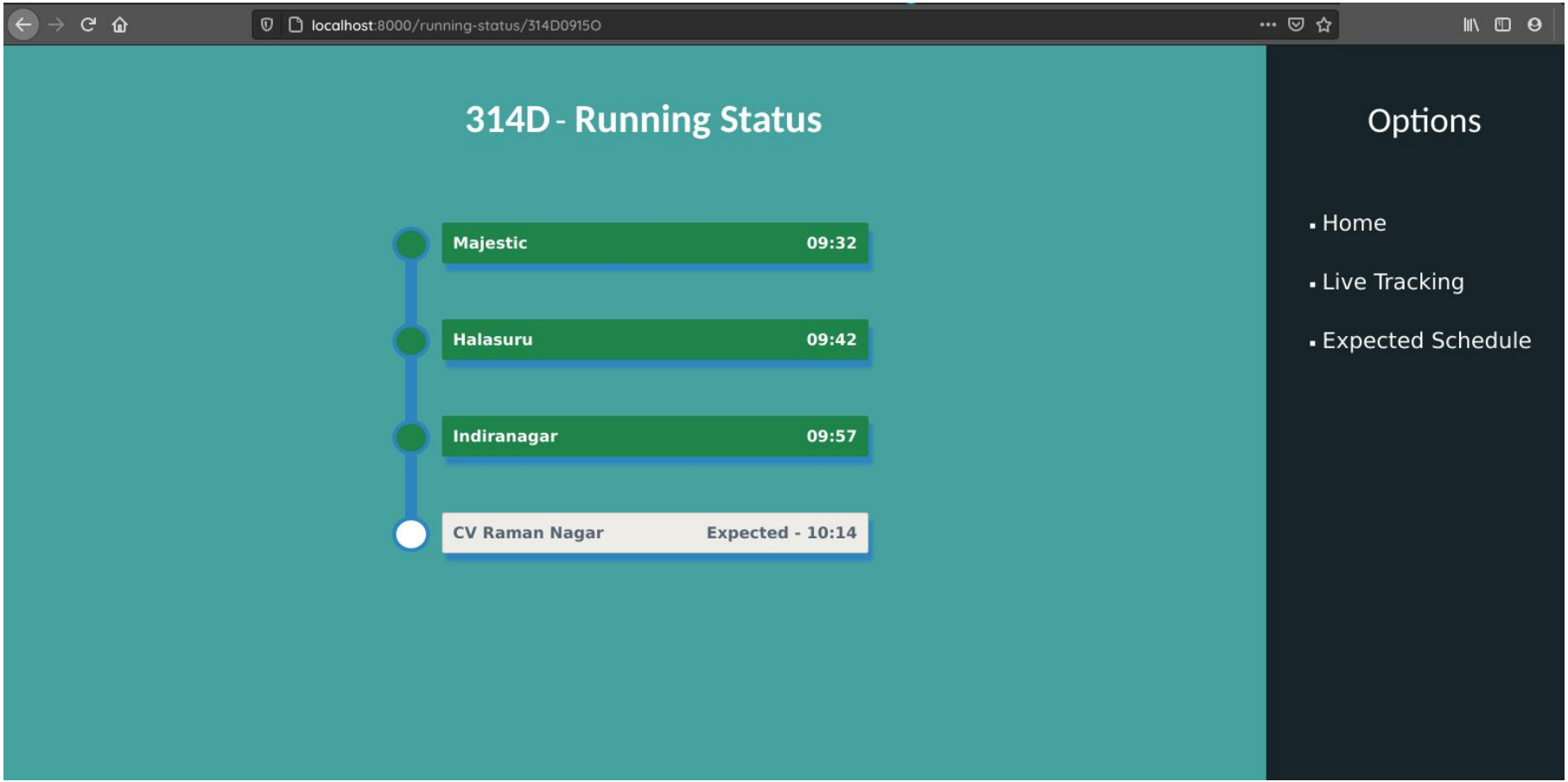
Track & Plan Commutes Online

- Built with **LAMP** Server Configuration
- Uses **MVC** architecture, through **Laravel**
- **Lightweight design** to facilitate heavy request traffic
- Will be hosted on a Raspberry Pi Zero for demonstration, after integration

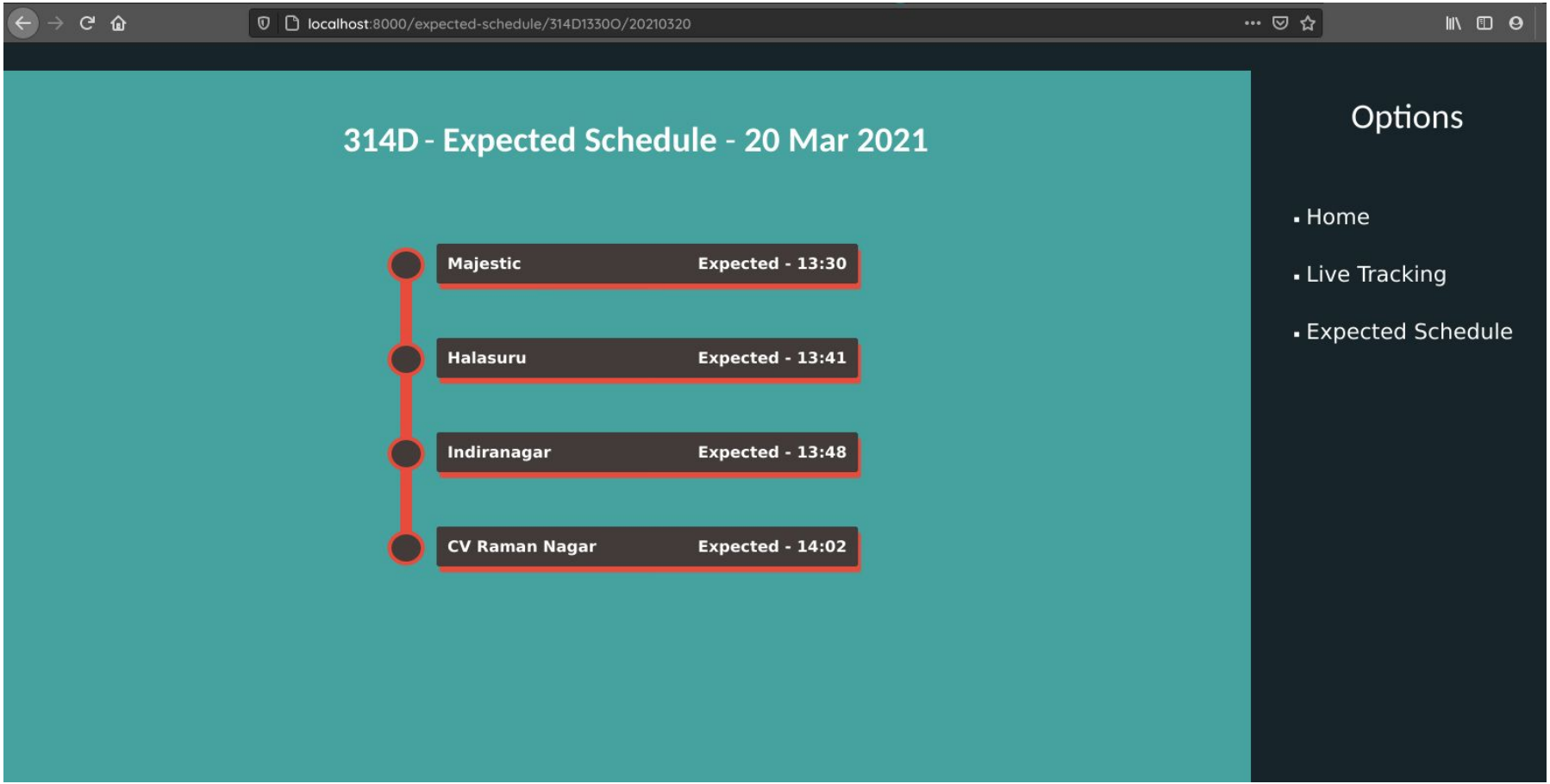
Search Buses between Stops . . .



Live Running Status . . .

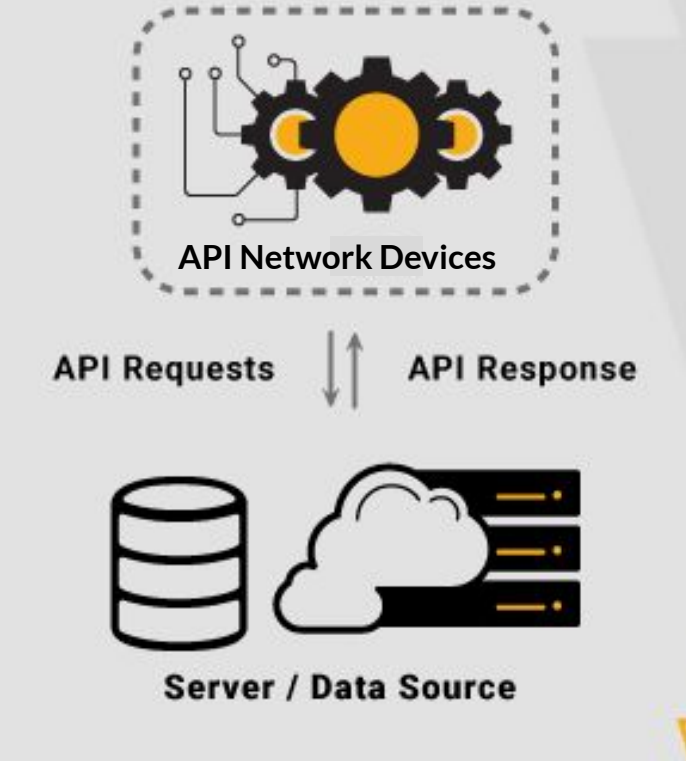


Tentative Schedule (on any day) . . .



API Server

Communication-Channel for the Transit System



- **Publish-Subscribe** architecture
- **API Endpoints** to store & retrieve data in/from **Remote-DB**
- **Authenticated** using **API-Tokens** that require regular **renewal**
- **JSON Request-Response** for lightweight IoT - DB communication
- 2 Client-Types: **Bus-Stops** and **Users**
- Extensible to token-wise access permissions

Generate API-Tokens . . .

JSON Request

POST localhost:8000/api/api-token

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2     "client_type" : "bus_stop",
3     "password" : "locatemybus_admin_access",
4     "stop_id" : "HLS0271"
5 }
```

JSON Response

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2     "Success": "API Token Generated",
3     "API Token": "0ViwhJCV29diTvDh6SmNk1FQP2ZK0g76U8Rc8KLZ1oP9GwIITOJijUM8sxNZ",
4     "Expiry": "2021-03-13"
5 }
```

Renew API-Tokens before Expiry . . .

JSON Request

PATCH localhost:8000/api/api-token

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "api_token" : "0ViwhJCV29diTvDh6SmNk1FQP2ZK0g76U8Rc8KLZ1oP9GwIITOJijUM8sxNZ",
    "stop_id" : "HLS0271",
    "client_type" : "bus_stop"
}
```

JSON Response

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "Success": "New Token Generated",
    "API Token": "Gd5mRA6319riBwYN5rSXOJcqs3XhPNKsp7yF5cTT5gGYAWCnaKAstnkxkJTU",
    "Expiry": "2021-03-13"
}
```

Log Arrival-Time of Bus . . . (PUBLISH)

JSON Request

POST localhost:8000/api/log-time

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "api_token" : "Gd5mRA6319riBwYN5rSXOJcqs3XhPNKsp7yF5cTT5gGYAWCnaKAstnkxkJTU",
    "stop_id" : "INR0371",
    "client_type" : "bus_stop",
    "rf_uid" : "60891530"
}
```

JSON Response

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "Success": "Trip Time Logged",
    "Time": 09:57,
    "Date": "2021-03-11"
}
```

Prevent duplicate logging . . .

JSON Request

POST localhost:8000/api/log-time

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "api_token" : "Gd5mRA6319riBwYN5rSXOJcqs3XhPNKsp7yF5cTT5gGYAWCnaKAstnkxkJTU",
    "stop_id" : "INR0371",
    "client_type" : "bus_stop",
    "rf_uid" : "60891530"
}
```

JSON Response

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "Error": "This trip was already logged"
}
```

Read next bus & predicted time . . . (SUBSCRIBE)

JSON Request

POST localhost:8000/api/get-arrivals

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "api_token" : "Gd5mRA6319riBwYN5rSXOJcqs3XhPNKsp7yF5cTT5gGYAWCnaKAstnkxkJTU",
    "stop_id" : "HLS0271",
    "client_type" : "bus_stop",
    "trip_id" : "314D13300"
}
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

JSON Response

```
{
    "Success": "Arrival List Ready",
    "Arrivals": [
        [
            "314D",
            "13:42:00"
        ]
    ]
}
```

Suitable Format for Display IoT

User APIs for authorized Third-Party apps . . .

JSON Request

POST localhost:8000/api/api-token

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2     "email" : "admin@locatemybus.com",
3     "client_type" : "user",
4     "password" : "locatemybus_admin_access"
5 }
```

JSON Response

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2     "Success": "API Token Generated",
3     "API Token": "LL6DlKw6bHtNo5KiJSjHkJdE9hgkwlqyesGhla024p04FRDpFTjfNhBibgM9",
4     "Expiry": "2021-03-13"
5 }
```

Demonstration of Use-Case

Bus Incharge Initiates Trip - Tripcode associates with Bus

NULL

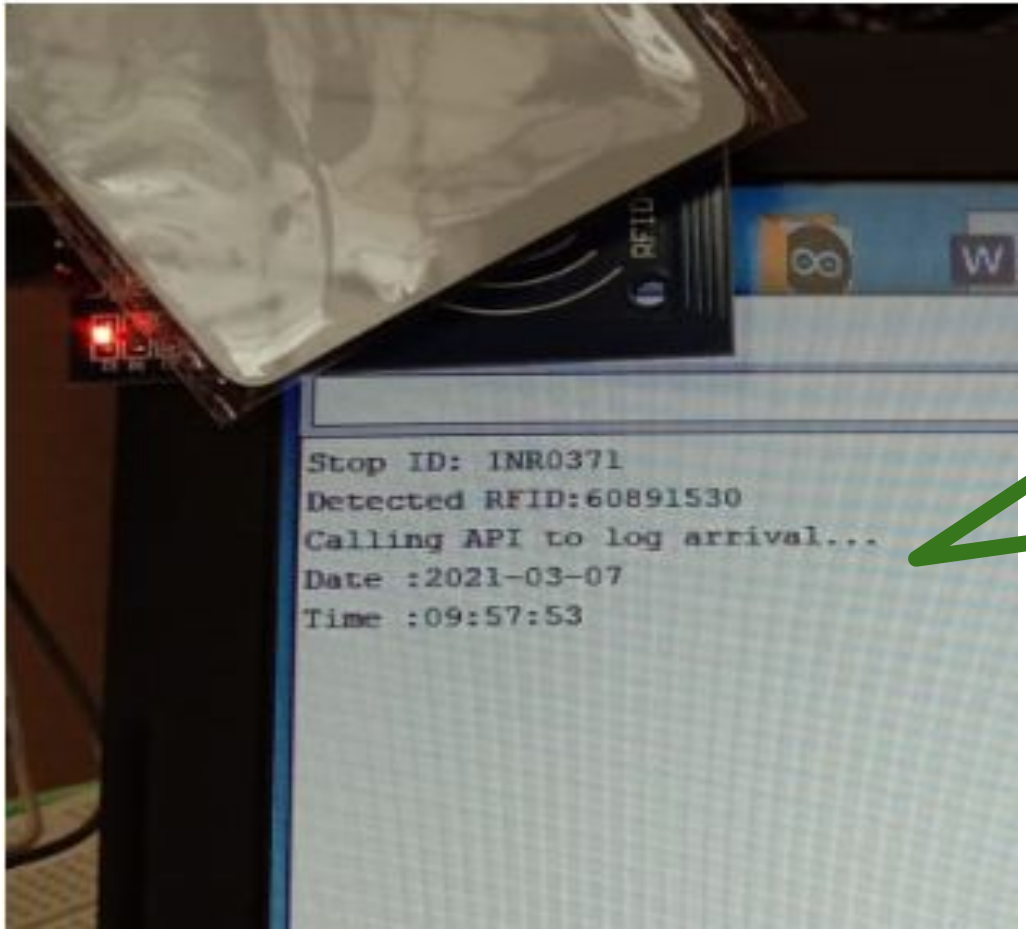
bus_id	rf_uid	model	variant	seating_capacity	class	current_tripcode
KA53ME6973	60891530	VOLVO	BF4	50	AC	
JJ87HQ0923	AAAD1286	TATA	DF554	50	AC	497U5524O
FV99RE9383	DF3232TA	MBENZ	MH545	45	NONAC	

BUSES TABLE

bus_id	rf_uid	model	variant	seating_capacity	class	current_tripcode
KA53ME6973	60891530	VOLVO	BF4	50	AC	314D0915O
JJ87HQ0923	AAAD1286	TATA	DF554	50	AC	497U5524O
FV99RE9383	DF3232TA	MBENZ	MH545	45	NONAC	

314D0915O

Bus arrives at Indiranagar - Beacon detects & scans bus RFID



Stop ID: INR0371
Detected RFID: 60891530
Calling API . . .
Date: 2021-03-07
Time: 09:57:53

Indiranagar sends API Request with **rf_uid** and **api_token**

JSON Request

POST localhost:8000/api/log-time

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2     "api_token" : "Gd5mRA6319riBwYN5rSXOJcqs3XhPNKsp7yF5cTT5gGYAWCnaKAstnkxkJTU",
3     "stop_id" : "INR0371",
4     "client_type" : "bus_stop",
5     "rf_uid" : "60891530"
6 }
```

JSON Response

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2     "Success": "Trip Time Logged",
3     "Time": 09:57,
4     "Date": "2021-03-11"
5 }
```

Server Updates Remote DB - Arrival of bus Logged at stop

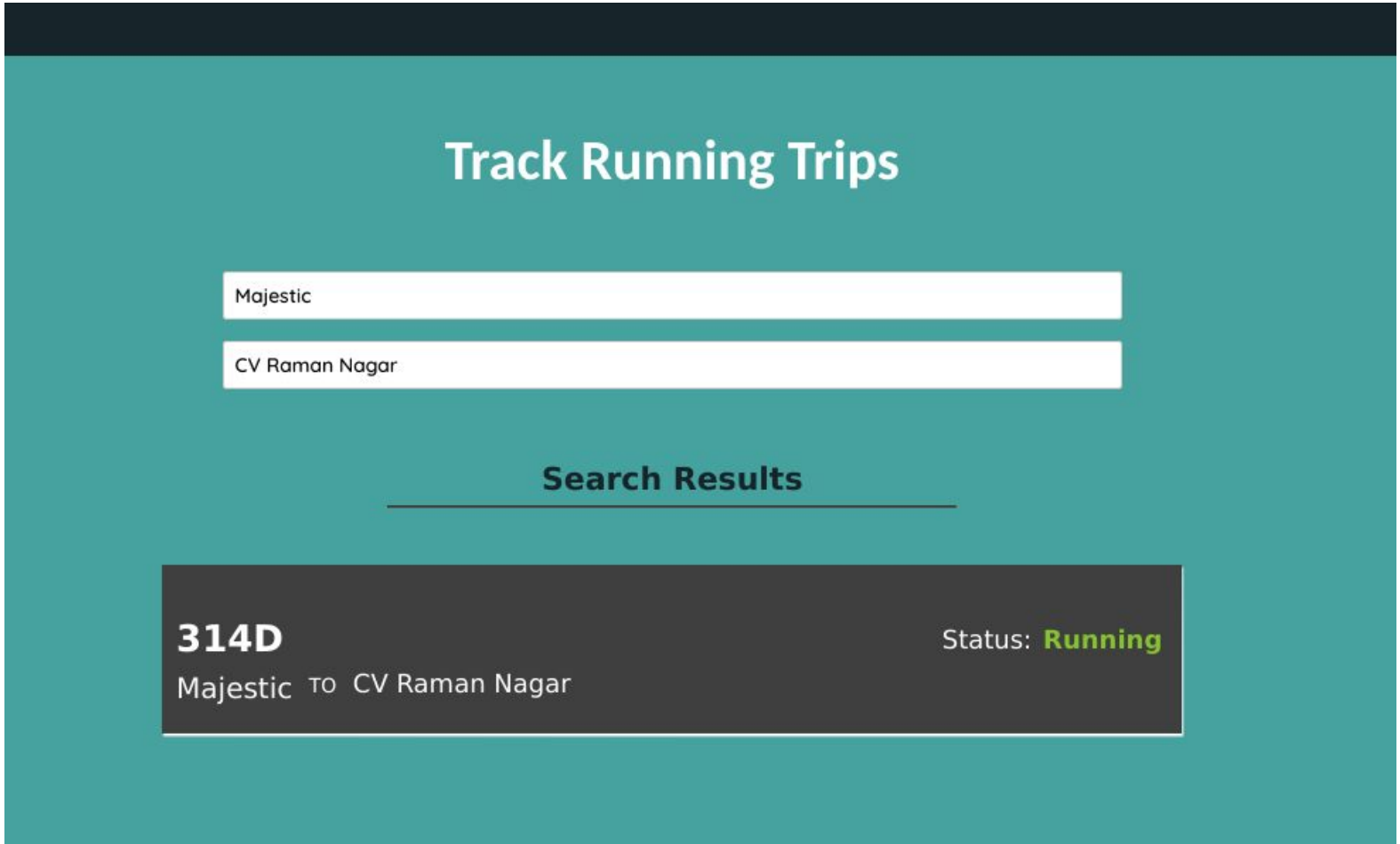
id	trip_id	stop_id	arrival_date	arrival_time
1	314D0915O	MAJ0091	2021-03-07T00:00:00.000Z	09:32:44
2	314D0915O	HLS0271	2021-03-07T00:00:00.000Z	09:42:53

TIME_LOGS TABLE

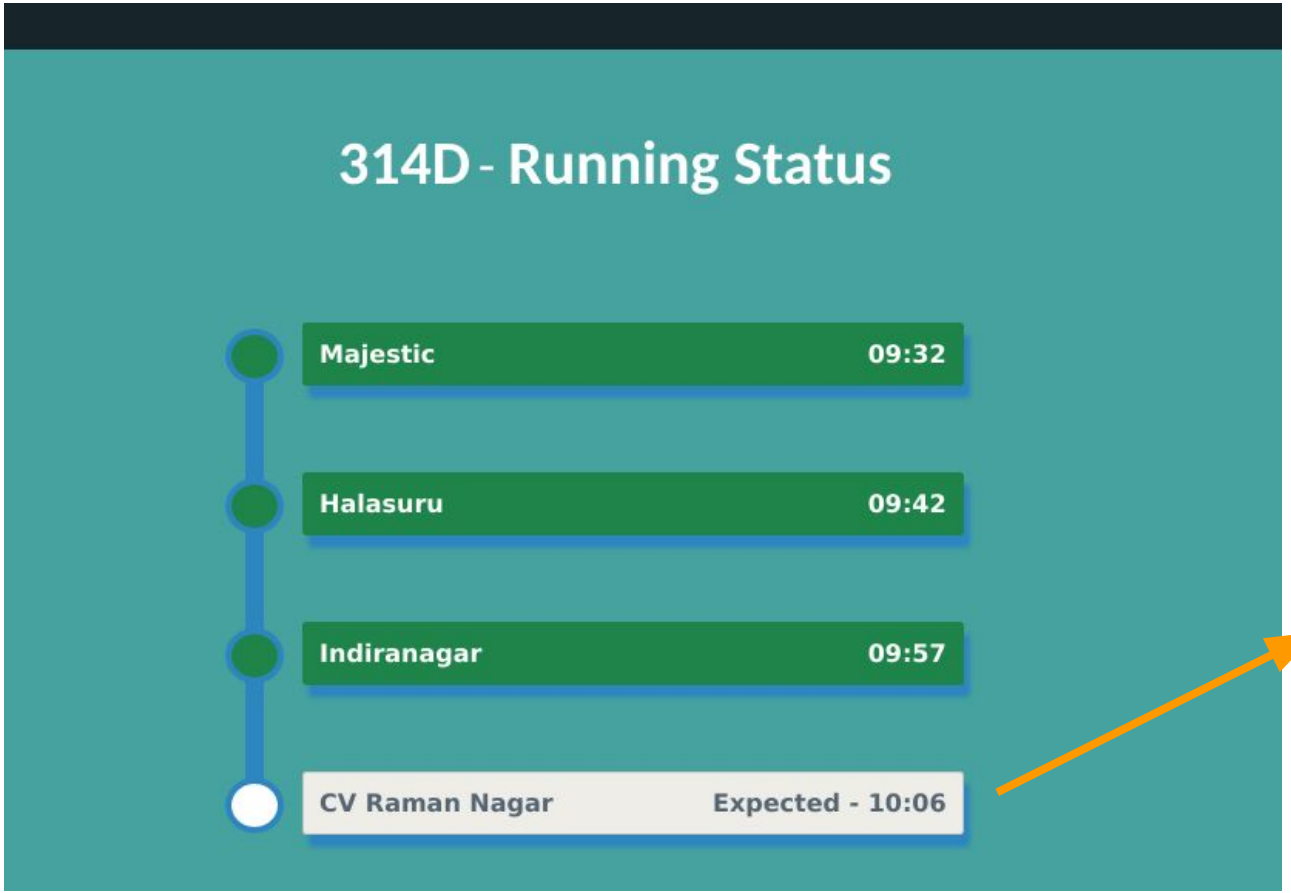
id	trip_id	stop_id	arrival_date	arrival_time
1	314D0915O	MAJ0091	2021-03-07T00:00:00.000Z	09:32:44
2	314D0915O	HLS0271	2021-03-07T00:00:00.000Z	09:42:53
3	314D0915O	INR0371	2021-03-07T00:00:00.000Z	09:57:53

NEW ROW

Commuter searches Running Trips - Website presents data



Commuter can Track Live Status - Website presents data



Will be predicted by ML Model after completion

Thank You