**Text

Description automatically generated**

**A group of people in a car

Description automatically generated**

**Smart Ride**

AIDI: Artificial Intelligence – Architecture, Design, and Implementation Ontario College graduate certificate program

Prof. Tareq Salim

December 14, 2021

Group 4:

200481194 - Anjani Sonavane

200487463 - Sanket Borkar

200494051 - Jeegniben Patel

200484628 - Rahman Armaghan

200481510 - Gurkirat Singh Saini

Contents List

Abstract……………………………………….…………………………………………… 3

Acknowledgments……………………….….……………………………………………. 4

Introduction/background………………………………………………………………….5

Methodology……………………………………………………………………………….6

Conclusion………………………………………………………………………………… 8

References and Bibliography…………………………………………………………. 22

List of Tables/Figures

Figure 1 – MSMD approach.…………………………………………………………….8

Figure 2 - A Generic View of the Characteristics Matching Layer…………………10

Figure 3 - Three Types of Rider Matching……………………………………………11

Figure 4 – Elbow graph…………………………………………………………………15

Figure 5 – Selecting a project in Google Cloud………………………………………18

Figure 6 – Google Cloud Platform Dashboard……………………………………….19

Figure 7 – Login Screen………………………………………………………………. 19

Figure 8 – Create new account………………………………………………………. 20

Figure 9 – Selecting an option…………………………………………………………20

Figure 10 - Add from and To Locations………………………………………………21

Figure 11 – To Location………………………………………………………………. 21

Figure 12 – Rider Questions…………………………………………………………. 21

Figure 13 – Driver Questions………………………………………………………….22

Figure 14 – Driver Dashboard…………………………………………………………22

Figure 15 - Chat Bot……………………………………………………………………23

Figure 16 – Jira board timeline………………………………………………………. 24

**Abstract**

Due to the continuous growth of autonomous vehicles, many countries started facing challenges related to environmental pollution, traffic congestion, not enough parking, accidents, as well as other inevitable problems. For example, a car with the ability to carry 5 people in full capacity, which is used to travel by and only for one person, 80% of the car capacity and fuel is wasted during individual travel. The best solution to overcome from all these problems is “Smart Ride, Car Sharing”.

In ride sharing, people who are going to their college/office and have almost similar timeslot and same route such that they can share a car, which is owned by a driver. In short, we share empty vehicle seats with people (known or unknown but verified).

There are few AI algorithms that we can use to find the shortest route and best path to travel. The driver shares their own car with people, who have the same or on the way source and destination. By smart sharing, we can reduce the traffic, pollution, less time to travel and most important cost effective for both driver/owner and rider. Smart ride would be beneficial for both driver and rider as they both get paid.

**Acknowledgment**

We would like to express our sincere gratitude to several individuals for supporting us throughout our Graduate study. First, I wish to express my sincere gratitude to my supervisor, Professor Tareq Salim, for his enthusiasm, patience, insightful comments, helpful information, practical advice, and unceasing ideas that have always helped us tremendously in our project development.

His immense knowledge, profound experience and professional expertise in the field has enabled us to complete this project successfully. Without his support and guidance, this project would not have been possible. I could not have imagined having a better supervisor in my study.

We would also wish to express our sincere thanks to the [Georgian College](https://www.rug.nl/) for accepting us into the graduate program and helping us achieve our dreams.

**Introduction/Background**

As it is mentioned in the abstract about the rising problems of pollution and parking spaces in the cities, people can use the public transport as well for the transportation. The major problems with the public transport are cost and timing. If a person opts for public transportation, it takes a longer time to reach the destination. Sometimes they must walk a lot to go to the bus stop/railway station and switch the transportation medium in the route. Likewise, if they choose cab, the cost will go beyond a normal limit. Moreover, for the daily transportation, the cab option is not feasible for some people, such as students

So, after seeing these problems we came up with an idea of ride sharing app. By smart sharing, we can reduce the traffic, pollution, time to travel and improve cost effectiveness for both driver/owner and rider.

Smart ride would be beneficial for both driver and rider as they both get paid.

**Methodology**

After investigating this real-world problem which is not limited to city or country but is an issue for the whole world. Even the most developed countries in the world are facing parking issues and dealing with the pollution problems. Countries are building their infrastructure widening their highways but of no use because of the traffic. At the same time rising prices of gas are also a big issue.

Therefore, we started to look up for the possible solutions to overcome this serious problem and we finally came up with idea of making a “smart AI based Carpool application” with the intention of not just to deal with above mentioned problems but at the same time helping the rider and driver from the financial point of view.

After selecting this topic, we started collecting all the information from the past similar kind of applications which were already in the market like Uber, Lyft etc.

We studied them deeply and concluded all the positive and negative aspects of the applications and finally came up with an idea of AI based approach to match the drivers and riders’ preferences at the same time with the help of feedback. We thought of giving the best experience to both the customer and the owner.

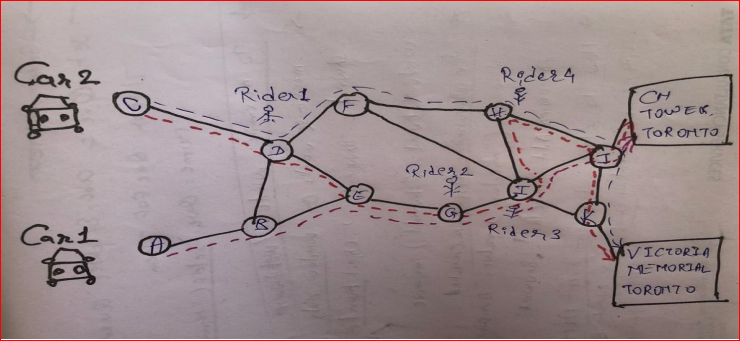
After studying all the research papers, we moved forward and have started developing “Smart Ride” application with the Django framework using Google Maps API in PyCharm. We planned to use the Python Google Maps API to make calls to different APIs using Python.

We planned to implement two types of users:

* Driver - A user who wants to share the ride with other people along the same route is a driver.
* Rider - A user (other than driver) sharing a ride. A user can book a ride and share a ride with driver.

**Conclusion**

The search for a suitable Machine Learning classifier led to the training and testing of feedback datasets with classifiers like Logistic Regression, K-Nearest Neighbors (KNN) classifier, Naive Bayes Multinomial classifier, Random Forests classifier, Neural Networks, and Support Vector Machine (SVM).

Out of these, we have tested on KNN and after implementation on feedback system, we tested on other algorithms also to check which gives the best result. Figure 1 – MSMD approach

Diagram

Description automatically generated with medium confidenceThe below Flow diagram describes execution steps of system. The execution starts from finding/showing rider request and follows by allocating a driver, executing matching layers, recording rider feedback, and computing two main characteristics. The final step is to predict two main characteristics based on trained and tested Machine Learning classifier for newly registering riders.

The Ride sharing model will include technology with two matching layers.

Figure 2 - A Generic View of the Characteristics Matching Layer

The model begins with rider registration where users provide required profile data along with specific details which can include positive digits on a scale of 1 to 5. The rider/driver selected in system is talkative, friendly, safe, on time, and comfortable. Once a driver/rider register in system, proposed model searches, and matches riders having a similar set of characteristics. The matching of riders using these characteristics shows on priority.

Diagram

Description automatically generated with medium confidence

Figure 3 - Three Types of Rider Matching.

The travelling time between locations of broadcasting rider and other riders is computed using the Google Maps APIs and verified if calculated time is less than trip time expected. If riders satisfy matching layer conditions, the system adds them onto the final trip itinerary, marking completion of trip formation.

After the trip completion, a user can see the feedback system begins where riders rate the driver as well as other riders on the trip. The feedback given by a user forms an essential dataset as the system uses feedback data to compute two main characteristics for every user. The determined characteristics are later employed by Machine Learning algorithms to predict better rider recommendations.

The two determined characteristics are Feedback-Given-Characteristic and Feedback-Received-Characteristic. Feedback-Given-Characteristic is derived based on feedback rider gives to other riders, while Feedback-Received-Characteristic is computed based on feedback rider gets from other riders. Two main characteristics are used to determine characteristics a rider most focuses on a trip while rating other riders. In the end, based on feedback patterns in past trips, system assigns the two most favoured characteristics to every rider.

The computations for determining main characteristics of a rider are quite complex and tediously high. Thus, after recording enough trip and feedback records, we use Machine Learning classification algorithms or classifiers to predict main characteristics of a rider, which eliminates need for complex computations. Machine Learning (ML) is a technology where a system learns, and trains based on an existing dataset and predicts outputs for new input data. In case of the Ride Sharing model, the thesis employs the Support Vector Machine (SVM) classification algorithm. After appropriate training and testing, SVM classifier predicts two main characteristics of newly registering riders. Riders are recommended based on predicted main characteristics. PFB the Flow Diagram.

Rider Request

New Registered Driver/Riders  
Chatbot

Show nearest Driver Details

Predict Characteristics using ML Classifier

Search Riders based on entered characteristics

Find Riders based on Predictions

Filter Riders according to requirements

Complete Trip

No

Available Vehicle Seats==0

Yes

Start & Complete trip  
Get review  
using Chatbot

Predict the main characteristics

Train and Test Machine learning Classifier

Machine Learning Module Selection:

In Machine Learning algorithms, we have done the prediction of the two main characteristics of newly registering riders. There could be a little error due to the presence of the imbalanced feedback dataset. In the case of an imbalanced dataset, for similar inputs, different outputs may be recorded, creating uncertainties during predictions. We need to find the Machine Learning classification algorithm or a classifier that could appropriately fit an imbalanced dataset and give quality predictions.

We worked for a suitable Machine Learning classifier that can be used on the training and testing of feedback datasets with classifiers like Logistic Regression, K-Nearest Neighbours (KNN) classifier, Naive Bayes Multinomial classifier, Random Forests classifier, Neural Networks, and Support Vector Machine (SVM).This process proved futile as we tried to create the classes manually and we soon found out that any classification solution would not work well with the kind of uncategorized data that we were working with. We then chose to move on to clustering models as they work well with uncategorized data. So, we chose K-means clustering, an unsupervised machine learning algorithm used for clustering data. It aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centres or cluster centroid), serving as a prototype of the cluster.

K-means was preferred by us because our dataset was uncategorized. After fitting the data through the K-means model, we ended up with 7 clusters each of which had both drivers and users in them, and each cluster had users with similar attributes for the purposes of maximum compatibility. This way we were able to match multiple divers to the one driver and vice versa. This was a certain improvement since the classification models were only matching a single rider with a single driver.  
 We concluded 7 clusters by running the data iteratively through the k-means model with different numbers of clusters and calculating the inertia. This gives the below mentioned elbow graph which tells us the optimum number of clusters to use for our data.

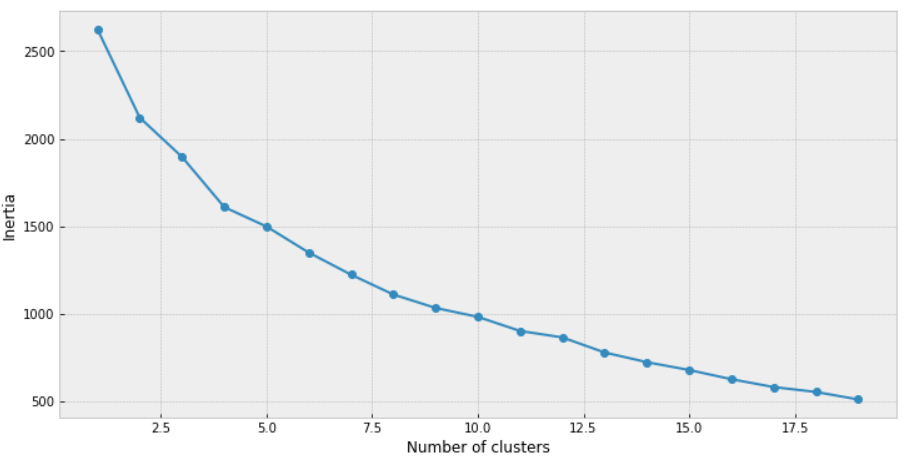


Figure 4 – Elbow graph

The ML algorithm works in the following manner. It checks the characteristics of the rider i.e., Talkative, Punctual, Music, Gender and Working and assigns a preference based on how close the values are compared to the driver’s preferences and vice-versa. The features Talkative and Punctual are assigned values ranging from 0 to 1 whereas Music, Gender and Working are also assigned either a 0 or 1. If a driver can find a suitable rider whose preferences are an exact match to the drivers, then the preference is set to 1. If there are minor differences between the preferences, then it is set to 2, else it is set to 3. Initially, we had taken only 1 rider in mind and designed our algorithm based on it. Now, with the implementation of the K-means model, it can match multiple users from the same cluster at once. Also, once the driver or rider registers their preferences on the website, their corresponding entries will go into the database. We can later fetch these records when drivers or riders will search for their preferred people and auto-assign a suitable preference based on the match between the preferences of both the parties.

Building Training Testing the Model:

We finally implemented a pilot application for user-friendly experience. The application provides functionalities like rider registration, broadcasting a request, completing a trip, and rating other users. As of now, we are not included billing part in this model, but we will try to cover predictions based on the factors such as travel distance and others, as mentioned above.

We have used an Agile software development life cycle where requirements, programming, and testing are often done concurrently. Please find below the planned steps:

1) We tried to meet all the requirements and develop the model as per desire.

2) Respond correctly to all kinds of inputs.

* Self – Review
* Peer – Review
* Unit Testing

3) Perform its functional requirement within an acceptable time.

* Integration testing
* Regression testing

1. Planned Visualization and Creation of Issues:
2. The visualizer supports the following functionalities:

* Create ride sharing scenarios.
* Solve ride sharing scenarios.
* Add adaptation issues (blocked streets) by clicking on a street.
* Visualize the solutions step by step.
* Visualize the overall distance traversed by carpools and passengers.

1. The following list shows what each of the navigation buttons does:

* Play - each of step in the plan is automatically displayed one after the other. Each state is shown for one second.
* Pause - stops the automatic displaying of the plan.
* Next - shows the next state in the plan.
* Restart - resets the visualizer to the first state (blocked streets are kept).
* Generate scenario - generates a new carpooling scenario.
* Send current state - the current state of the scenario is sent to the solver to get an updated solution, e.g., to resolve new adaptation issues (blocked streets).

**Output of Work:**

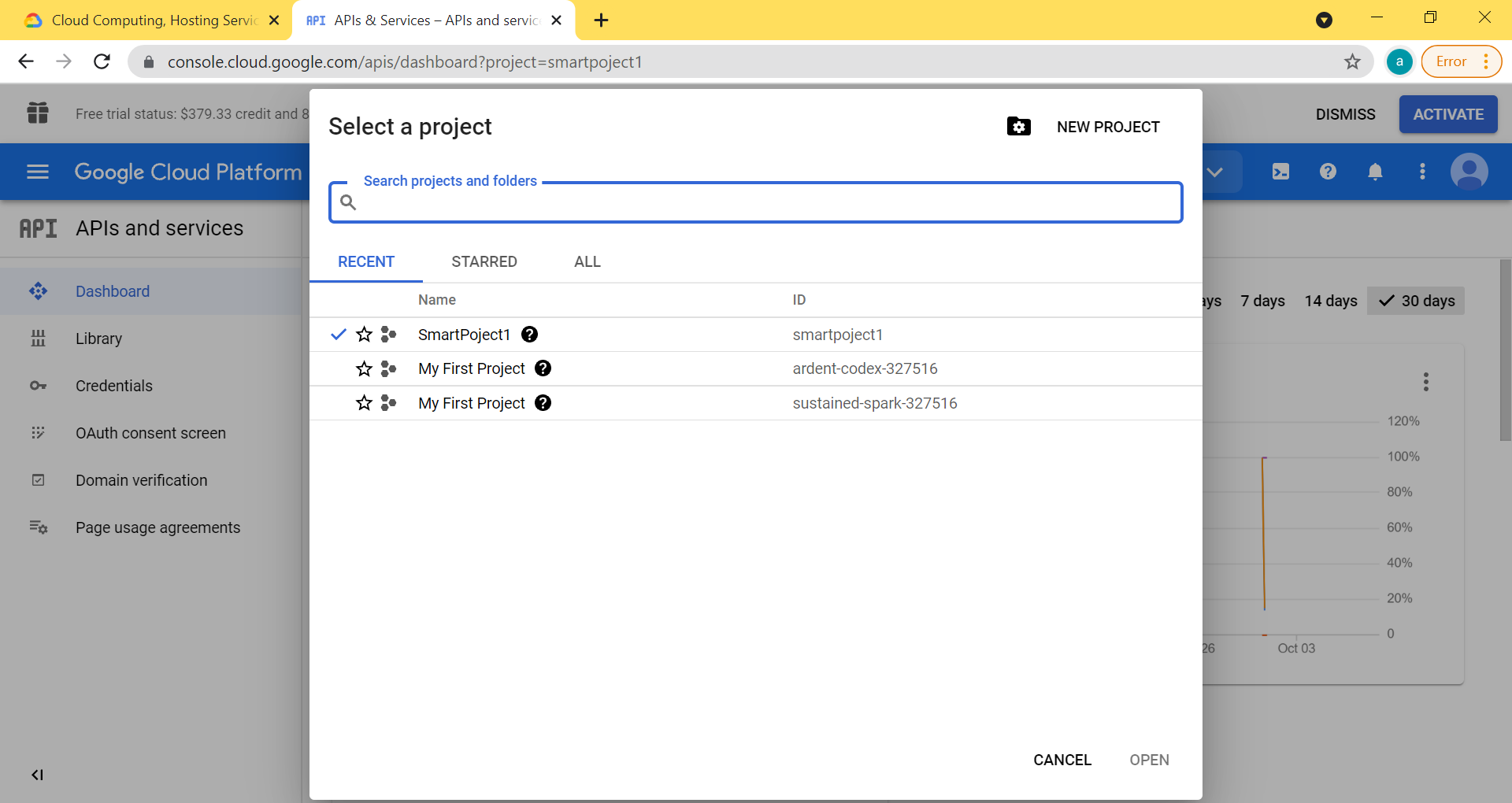
****

Figure 5 – Selecting a project in Google Cloud

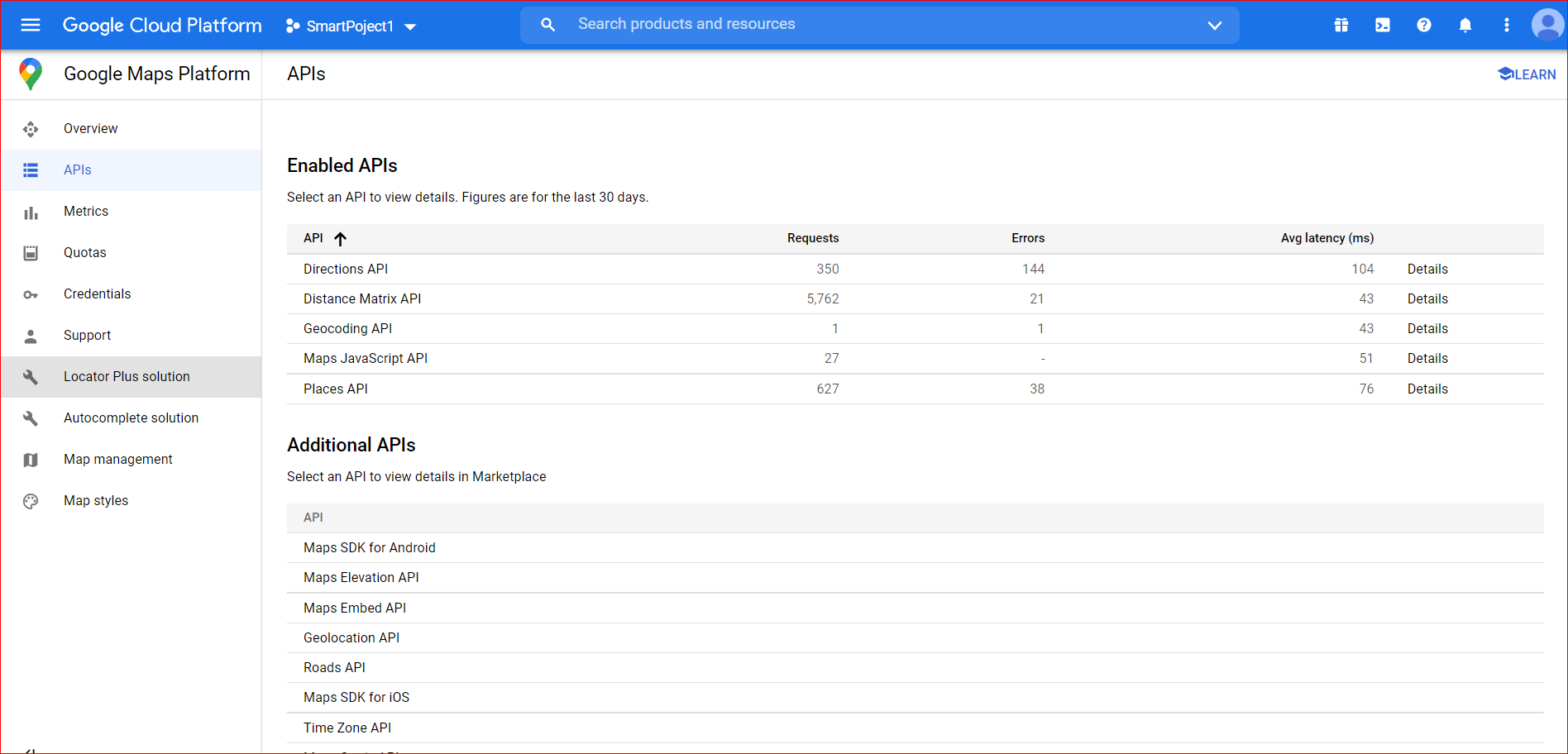
****

Figure 6 – Google Cloud Platform Dashboard

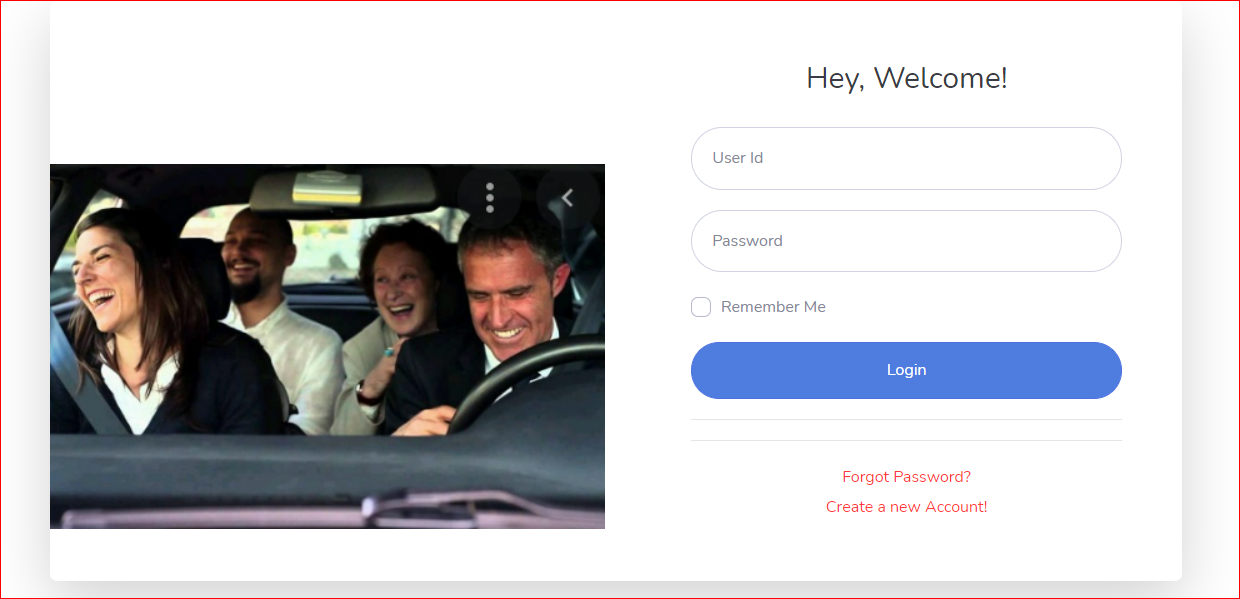
****

Figure 7 – Login Screen

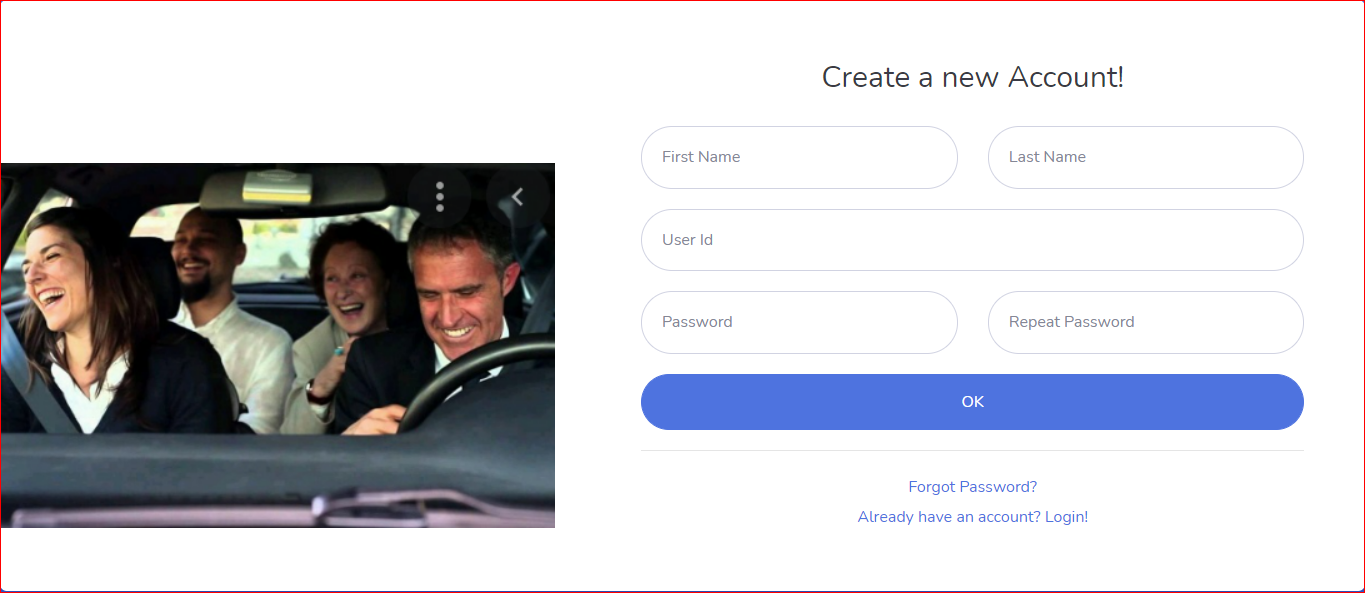
****

Figure 8 – Create new account

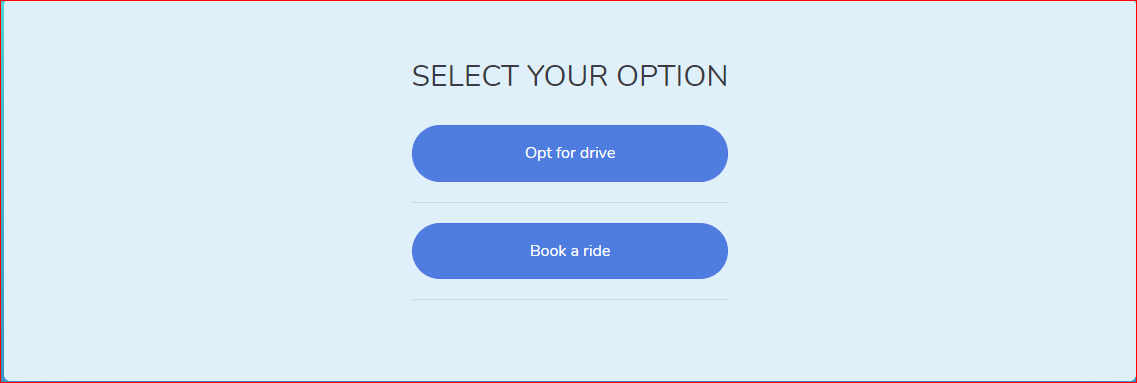
****

Figure 9 – Selecting an option

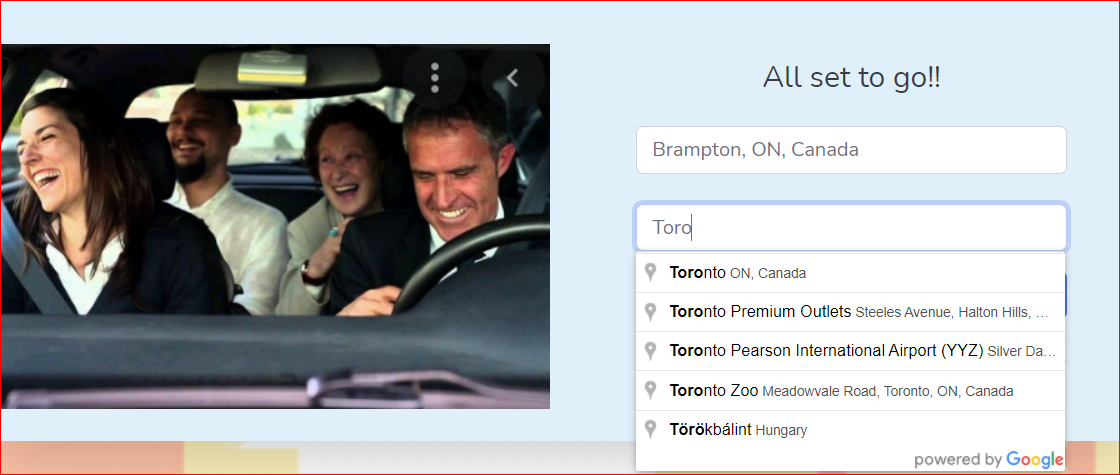
****

Figure 10 - Add from and To Locations

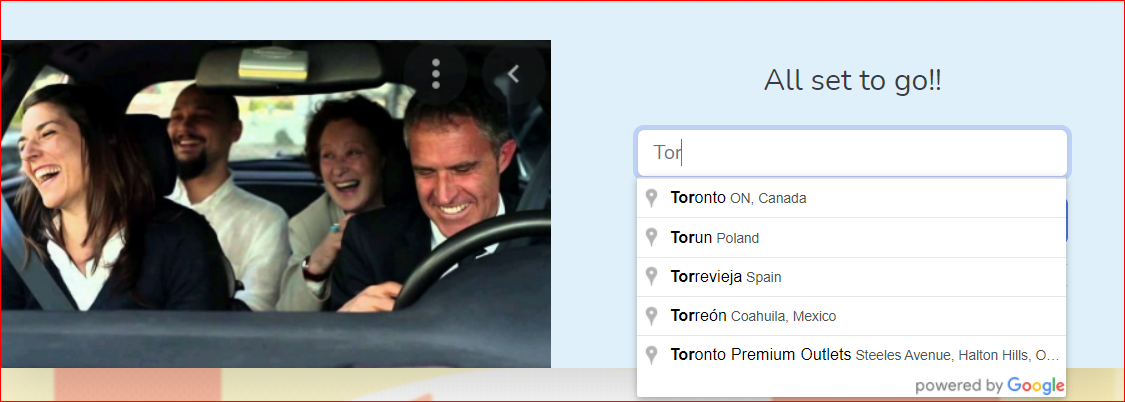
****

Figure 11 – To Location

**Graphical user interface, text, application, Word

Description automatically generated**

Figure 12 – Rider Questions

**Graphical user interface, text, application

Description automatically generated**

Figure 13 – Driver Questions

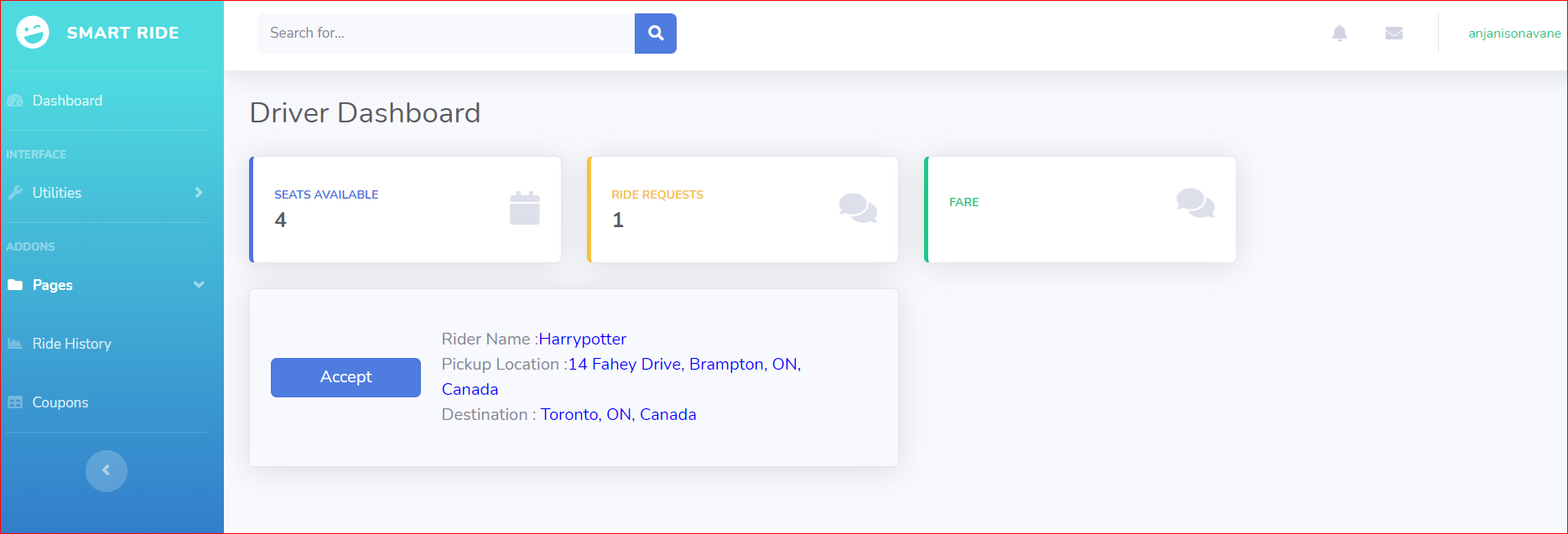
****

Figure 14 – Driver Dashboard

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Figure 15 - Chat Bot

**Timeline**

[SMAR board - Agile board - Jira (atlassian.net)](https://smartride-group4.atlassian.net/jira/software/projects/SMAR/boards/1)

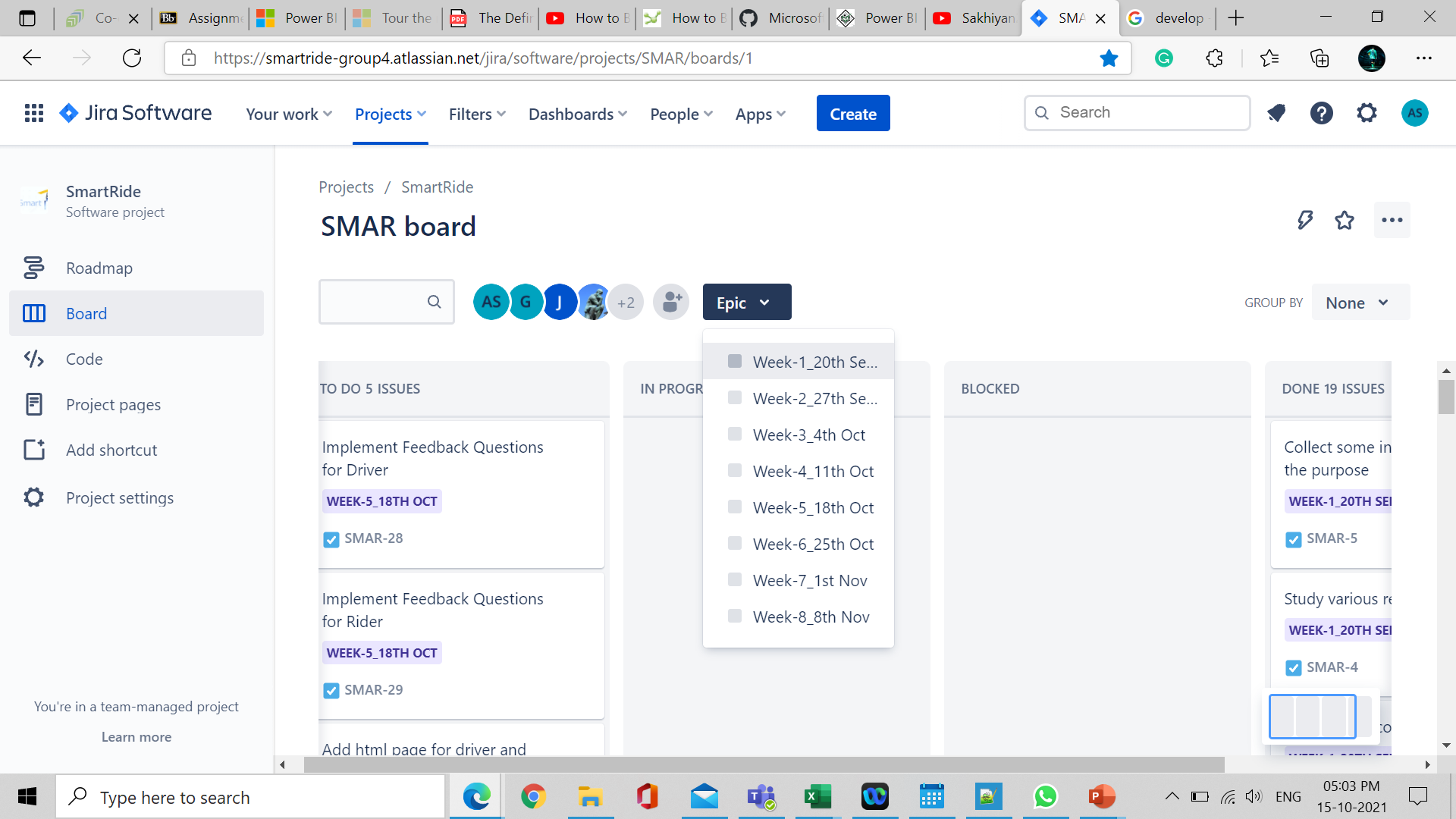
****

Figure 16 – Jira board timeline

**References and Bibliography**

[1] Ferguson, E. (1997) The Rise and Fall of the American Carpool: 1970-1990. Transportation, 24, 349-376. <https://doi.org/10.1023/A:1004928012320>

[2] ONU-Hábitat and ONU-Hábitad (2015) Reporte Nacional de Movilidad Urbana en México 2014-2015. Rep. Glob. en Asentam. Humanos, p. 100

[3] Ferguson, E. (1990) Demographics of Carpooling. Transportation Research Record, 1496, 142-150

[4] Minett, P. and Pearce, J. (2011) Estimating the Energy Consumption Impact of Casual Carpooling. Energies, 4, 126-139. https://doi.org/10.3390/en401012