CPS 630: Web Applications
Plan for Smart Services (P2S)

TEAM 16

Aitirja Chowdhury (500832196)
Felicia Levina (500856106)
Mehnaaz Rahman (500835150)

Percentage of tasks completed by:

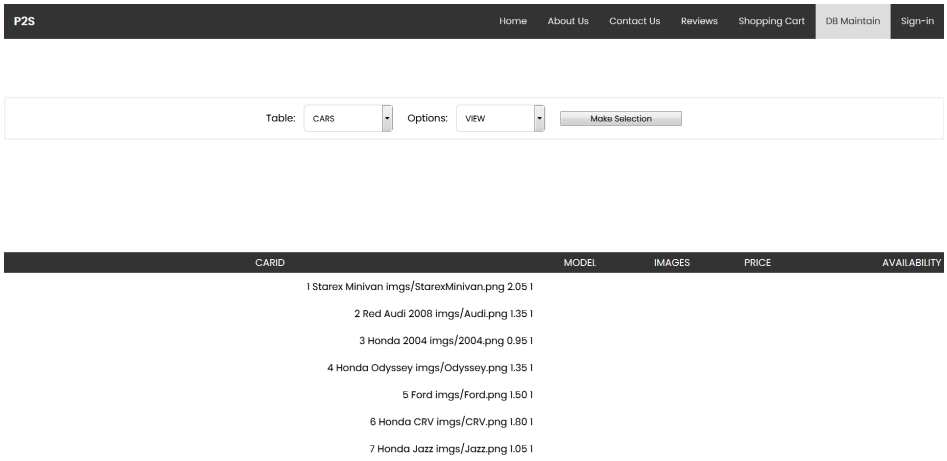
Aitirja Chowdhury: 33%	Felicia Levina: 33%	Mehnaaz Rahman: 33%

Project Objectives, Languages, Tools

Our goal for this project was to create a user-friendly application that allowed clients to plan eco-friendly trips throughout the city through the sharing of vehicles. By using the knowledge and tools provided to us in this course, through the form of various front-end and back-end technologies, Google Maps API, and more, we wanted to challenge ourselves and extend our knowledge of the course content to create an impressive project that could potentially come to fruition in the near future.

That said, for the design and implementation of our project, we used HTML5, CSS3, JavaScript, JQuery, PHP, MySQL, Bootstrap, and the Google Maps API, while incorporating the use of the Model-View-Controller pattern on the architecture of the overall system. While HTML5, CSS3, JavaScript, JQuery, and Bootstrap were used on nearly every page of the application to create a user-friendly and mobile-friendly interface, the Google Maps API was only used on the service pages so that the user could view and determine an address for their trip. PHP and MySQL were only used for the pages that required access to the database (i.e., the shopping cart page, sign-in page, sign-up page, the service pages).

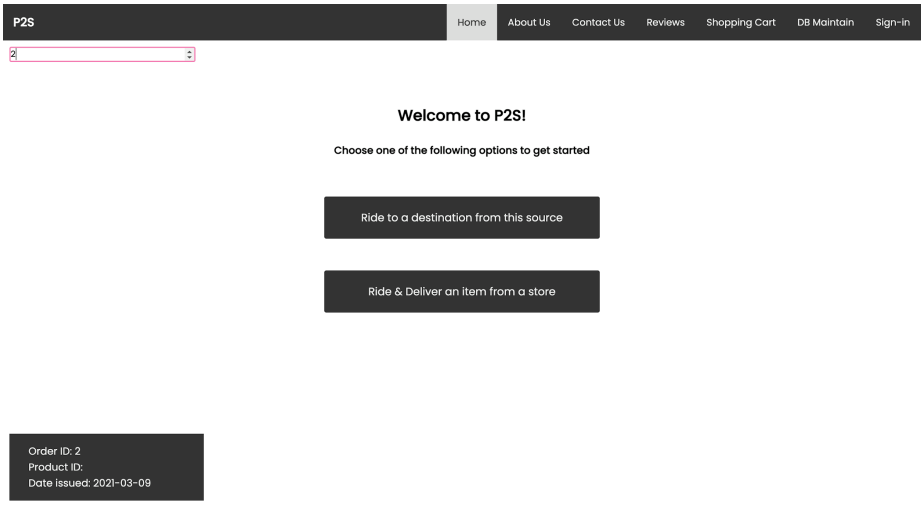
The database maintenance mode allows certain users to perform operations, such as viewing data, on the company's database. For instance, in the case of selecting the table cars from the page's drop-down list and view under options, the user can view all the information associated with cars in the company's database, as seen below.



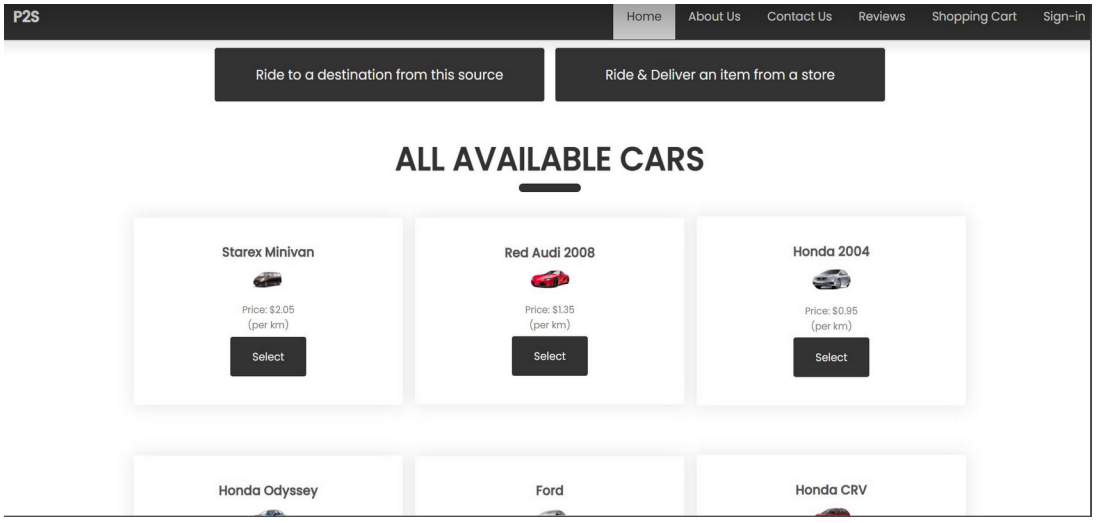
In order to implement the view mode, the following MySQL command was used:

SELECT * FROM cars

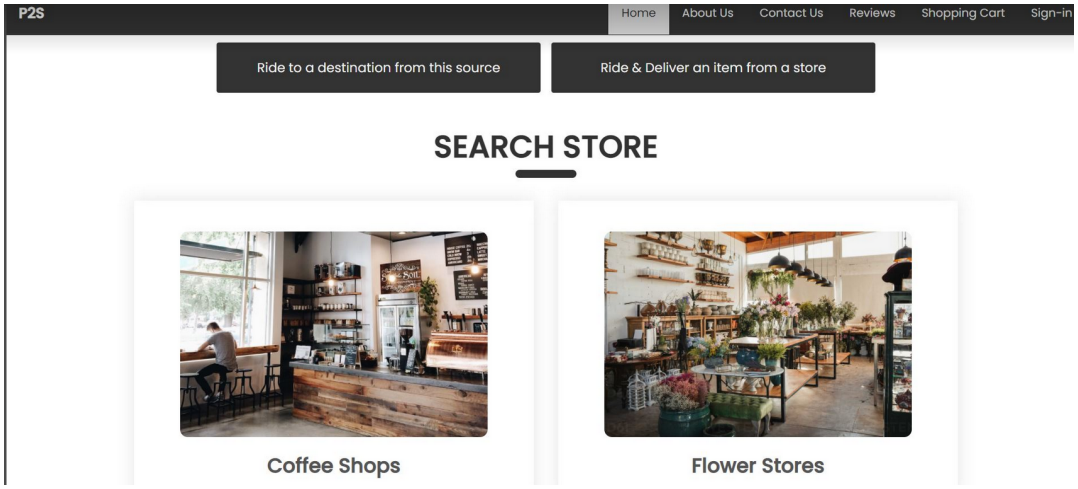
Application User Interfaces



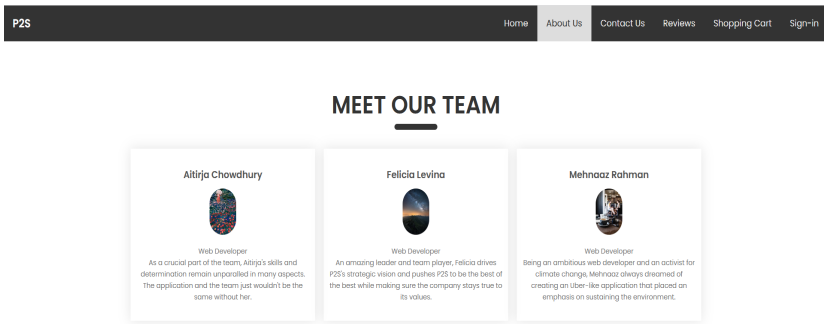
This page of the application is the main page that the user is directed to upon navigating the website. From here the user can navigate to the services pages by clicking on the “Ride to a destination from this source” button or the “Ride & Deliver an item from a store” button. If the user clicks on the first button, they will be navigated to the following page from where they can view and choose available cars by model and price, select the source and destination address for their trip, date and time of their trip, view the addresses using the Google Maps API, save orders, process payments, etc. At the same time, if the user would like to view their past orders they have made, they can do so by putting in their order id. This shows the order and when it was purchased. Of course, they would need to have an account to be able to use this service.



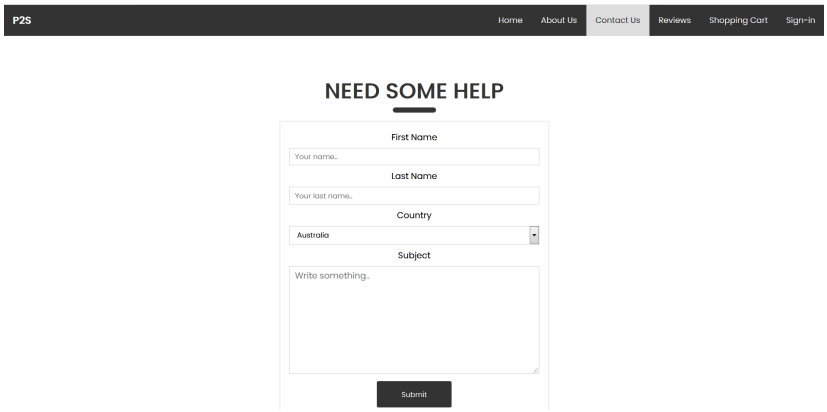
If the user clicks on the second button, “Ride & Deliver an item from a store”, they will be directed to another service page (as seen below) where they can view and choose an item by store and price, as well as select the source and destination address for their trip, date and time of their trip, view the addresses using the Google Maps API, save orders, and process payments similar to options provided by the first page.



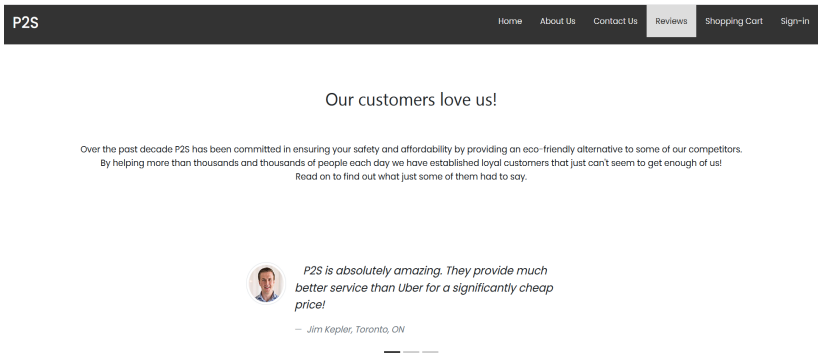
Alternatively, the user can also click on any of the links contained in the navigation bar located at the top of each page. However, the user should keep in mind that they need to first create an account before they can use any of the services.



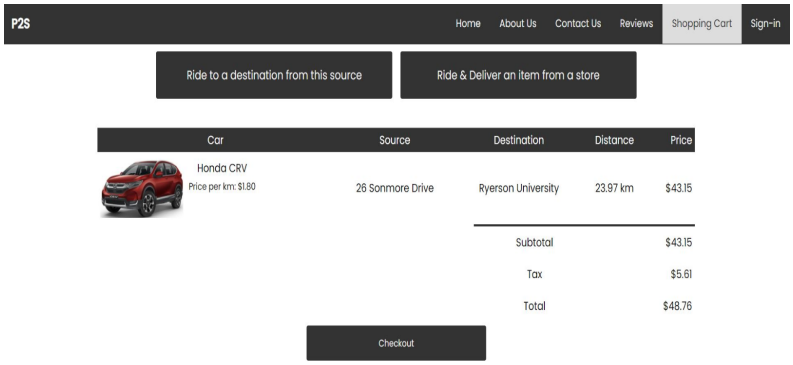
If the user clicks on the About Us page from the navigation bar, they will be redirected to a page where they can view some general information about each of our members.



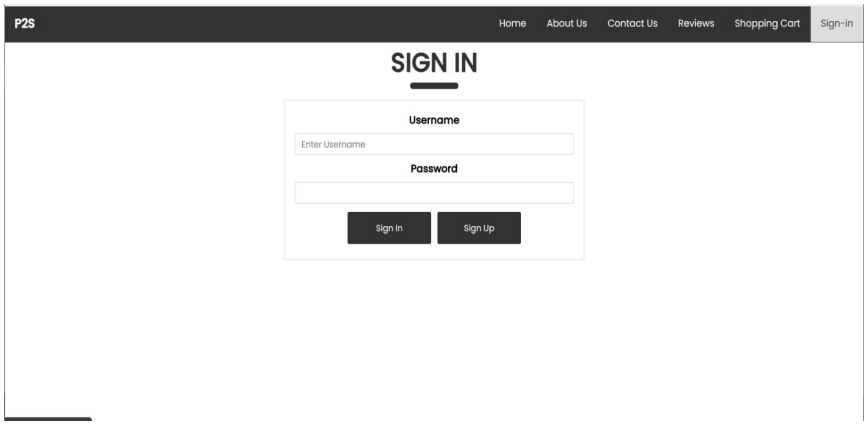
Upon reaching the Contact Us page, the user can fill out the fields in the form and click the submit button to send us any questions and concerns that they may have regarding either the navigation of our website or the services offered by our company.



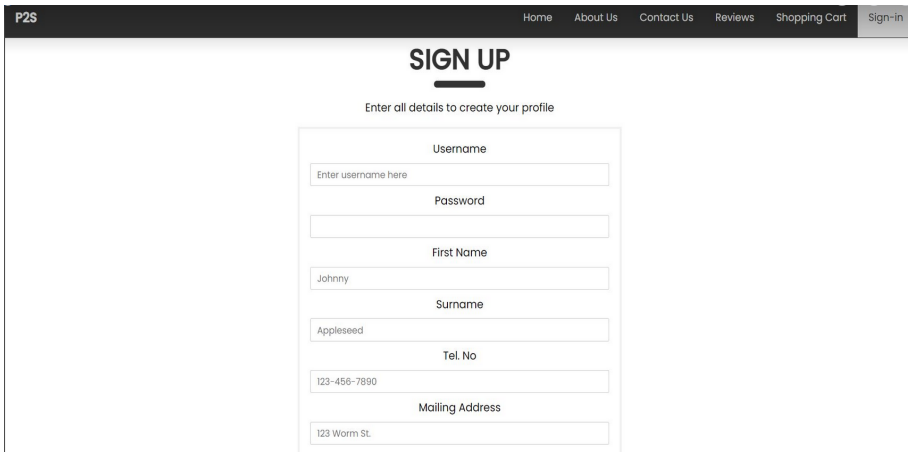
The Review page simply contains some favourable reviews and feedback by other “clients” who recommend our services.



The shopping cart page allows users to save orders through the drag and drop method, however, they must have an account and be signed in to do so. If not, they will be redirected to the sign-in page from where they can login to their account.



The sign-in page is used to enter and use the system. If the user does not fill in the correct username and password that corresponds to their account, they will receive an error message. However, if they enter the correct credentials, they will be redirected to the home page from where they can navigate to the services pages by clicking on the buttons centered on the page. In the case that the user does not have an account, they can click the link at the bottom of the page to create an account.



On the sign-up page, the user can create an account / profile by filling in all the fields, such as username, password, phone number, etc. If their password is shorter than 8 characters, or they

enter invalid characters in any of the fields, they will receive an error message telling them so. Upon the successful creation of an account, they will be redirected to the home page.

Project Design & Database Implementation

In our database, we have separate tables for order information (Order), user information (Users), trip information (Trips), vehicle information (Cars), and item information (Items, Flower, Coffee). The Users table contains the fields USERID, USERNAME, PSWRD, FIRST_NAME, LAST_NAME, TEL_NO, MAIL_ADDR, EMAIL, and BALANCE, where the USERID acts as the primary key. The Trips table contains the fields TRIPID, USERID, CARID, SRC, DEST, DIST, PRICE, and TM, where the TRIPID acts as the primary key, and the USERID and CARID act as foreign keys that are being referenced from the Users and Cars tables, respectively. The Cars table contains the fields carid, model, imgs, price, and available, with carid acting as a primary key. The Flower table contains the fields flowerid, flowerType, storeCode, img, and price, with flowerid acting as the primary key. The Coffee table contains the fields coffeeid, coffeeType, storeCode, img, and price, with coffeeid acting as a primary key. The Order table contains the fields ORDERID, USERID, TRIPID, PRODUCTID, and DATE_ISSUED, where the ORDERID acts as the primary key and USERID, TRIPID, and PRODUCTID act as the foreign / referenced keys. The Items table contains the fields PRODUCTID, USERID, SRC, DEST, DIST, and TM, with PRODUCTID as the table's primary key and USERID as the foreign key.

The following is the database schema diagram of our application: