**University of Central Missouri**

**Advance Systems Project**

**ShareNride Web application**

**Enrolled Under:**

Prof. Sam Ramanujan

**Submitted To:**

Prof.Narsimha Paravasthu

**Submitted by**

Ranga Rayudu Rayapudi

UCM id:700658520

Email:rxr85200@ucmo.edu

**Abstract:**

**Objective:**

ShareNRide websites enables students in University of central Missouri to have an efficient transport system in place. The overall mission is to enable students find reliable and cost-efficient transport from UCM to nearby places and from other places to UCM. The student community post their requests and availability of rides on this site. The admin maintains the routes and the prices per miles.

**Version History:**

|  |  |  |
| --- | --- | --- |
| **Version type** | **Date** | **Description** |
| Project Proposal | 6/10/17 | Project Proposal |
| Research on the technologies used | 8/10/2017 | Learning new technologies Angular 4, Bootstrap 3, PHP PDO  Required for project |
| Added Functionalities | 8/25/2017 | Created basic environment, database, created login, register and the home module. |
| Structuring the project | 9/1/2017 | Restructured database and implemented the use cases given in the project proposal |
| Adding additional functionalities | 10/15/2016 | Added the new functionalities, implemented backend connection, addedrouting concepts and Route guards |
| Documenting use cases, activity diagrams | 11/10/2017 | Added use case diagram description to the document, added activity diagrams |
| Added class diagrams, references. | 11/15/2017 | Included class diagrams, references |
| Final Documentation | 11/16/2017 | Included help log and added used SQL statements and resources and reference used. |
| Final Submission For grading | 12/17/2017 | Added few validations |

**Technologies Used:**

Front-end Framework: Angular 4

CSS: Bootstrap 4,

Backend : PHP-PDO

Database: MySQL, XAMPP- PhpMyAdmin

Build: Web pack

**Project Description:**

ShareNride is a solution for the existing student transportation problem in UCM. So far, we have UCM bus service from UCM main campus to Lee summit campus with limited available seats and requires a bus pass. Students use few social networking groups like WhatsApp, Facebook etc. to post and request rides. But the access to these group are limited. So, we need a system in place to resolve this issue.

The initiative behind this system is to have a common platform for the students or university staff to request or share their trip with UCM Mules. The routes are managed in the routes managed by the admin. UCM Mules can register as driver or passengers and join the available rides posted by drivers and even create a customized ride.

The system has 2 Main Users.

**Rider:**

Students without car can register as a rider and look for the available rides in the system. If they cannot find a ride at the time and date they are looking for, they can create a custom ride with origin and destination along with number of people riding along with them. These rides are then accepted by Drivers based on their availability. Once the ride is finished the ride history is added to his profile.

**Driver:**

UCM Mules (Students or faculty) with car can register their car. Once the rider logins he/she sees all the available. He can specifically see all the custom request rides and accepts the requested ride. Rider can also see his ride history. Rider can see also request for a ride and join as a user.

**Modules:**

* Register/Login
* Available Rides
* Manage Driver Car Details
* Post a ride
* Accept a ride
* Join Ride
* Request customized ride
* Start/End Date
* Buy coupons
* Ride History as Driver
* Rider History as Rider

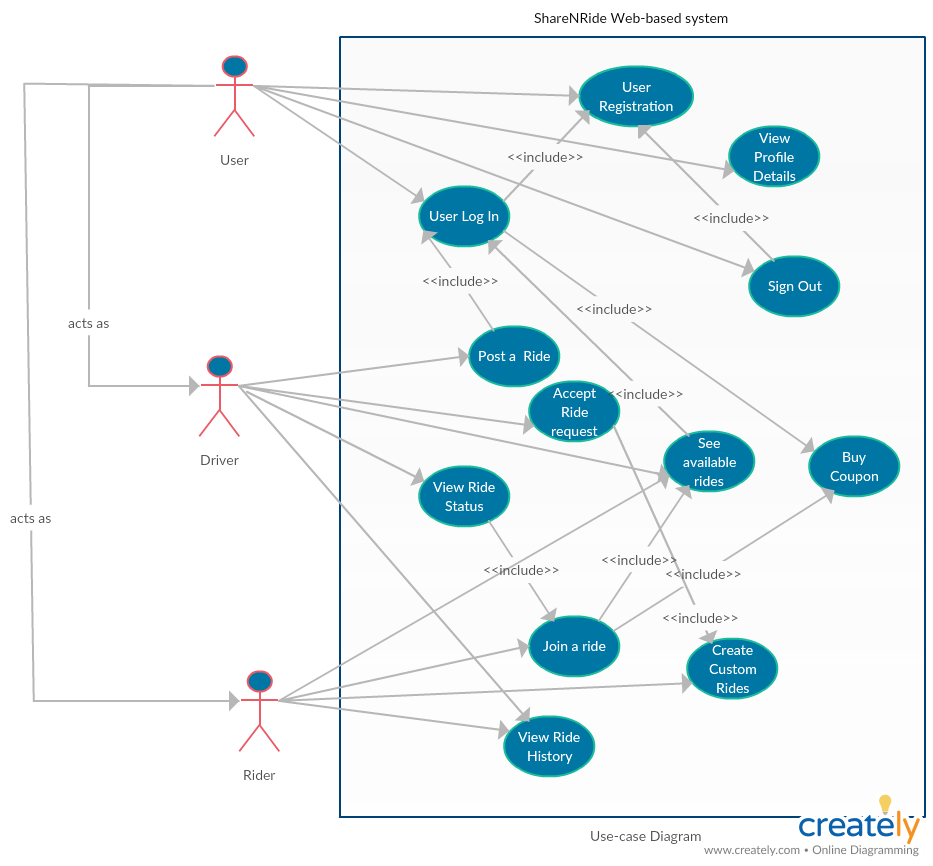
**Accessing the system:**

Any UCM Mule (Staff/Students) can register and access the System.

**Functionalities:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use case** | **Priority levels: 1:2:3** |
| **SNR21** | User Registration | **P1** |
| **SNR22** | User Login | **P2** |
| **SNR23** | Get Available Rides | **P2** |
| **SNR24** | Request a Ride | **P1** |
| **SNR25** | Post a Ride | **P2** |
| **SNR26** | Create Custom Ride | **P2** |
| **SNR27** | Join Ride | **P1** |
| **SNR28** | Manage User coupons | **P2** |
| **SNR29** | Accept a Ride | **P2** |
| **SNR30** | Start/Stop Ride | **P3** |
| **SNR31** | Driver Car Details | **P3** |
| **SNR32** | Ride History | **P2** |

**USE CASE DIAGRAM**



**Use Case Description:**

Share N Ride web based application use case Description.

**Use Case 1:User Registration**

* Needed for new Users

**Primary Actor**: Rider, Driver, Admin

**Scope:**  ShareNRide web based System

**Level:**  user level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** System in Production

**Minimal guarantees:** Able to register the User in Database

**Success guarantees:** Able to Login

**Trigger:** User Need /share a Ride

**Main success scenario**:

1. Successful Registered as a User in the system.

**Frequency of Use:** Moderate

**Priority**  High

**Use Case 2: Login**

User login to access the application features.

**Primary Actor**: Rider,Driver

**Scope:**  ShareNRide web based System

**Level:**  User level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** User must be registered in database

**Minimal guarantees:** Able to see the application Home page based on the credentials.

**Success guarantees:** User able to access the application features.

**Trigger:** Need for access to the application

**Main success scenario**:

1. User logins in with the credentials. System verifies the credentials and redirects to index page.
2. Error message is displayed if wrong credentials are entered and if not a registered user. System Prompts to register.

**Frequency of Use** High

**Priority**  P1

**Use Case 3: See Available Rides**

As the user login into the website the available rides are shown.

**Primary Actor**: Rider,Driver

**Scope:**  ShareNRide web based System

**Level:**  user level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** User is logged into the system

**Minimal guarantees:** User see all the available rides.

**Success guarantees:** Successful access to the use the system database.

**Trigger:** Logged in as a rider or Driver.

**Main success scenario**:

1. System navigates the user into home page.
2. System queries the database and selects all the rides posted.
3. If login fails the user is asked to enter a valid credentials.
4. Once logged user can Join Rides Available.
5. User can apply if he has coupons available in his account.
6. Joined ride message is displayed on success.

**Extensions: 1.** If no available rides Message is Displayed.

1. User can Request ride by posting or create a customized ride.

**Frequency of Use High**

**Priority**  P1

**Use Case 4: Request a custom ride**

**Primary Actor**: Rider

**Scope:**  ShareNride web based System

**Level:**  user level

**Stakeholder’s and interests**: ShareNride stakeholders.

**Precondition:** User logged into the system.

No suitable rides available at required time.

**Minimal guarantees:** Request Is made successfully. Inserted in the rides table.

**Success guarantees:** Ride request is added to the custom ride requests.

**Trigger:** Rider needs a ride and available rides does not meet the requirement.

**Main success scenario**:

1. Rider logs into the system and search for suitable rides
2. User enters the origin & destinations and time and request rides.
3. Rider Enter s all the required details and request a ride.
4. Success message is displayed.

**Extensions:** If valid ride detail are not entered Error messages are displayed.

**Frequency of Use:** High

**Priority** Moderate

**Use Case 5: Post a ride**

**Primary Actor**: Rider

**Scope:**  ShareNRide web based System

**Level:**  User level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** Driver logged into the system

**Minimal guarantees:** Ride gets inserted into the database.

**Success guarantees:** Posted message is displayed .

**Trigger:** Driver wants to share a ride.

**Main success scenario**:

1. Driver login into the system.
2. Driver enters the ride details and clicks post ride button.
3. Ride is inserted into the table.
4. Posted Ride is visible to the users.

**Extension**: **1.**  If valid details are not provided error message is displayed.

2. If the date of the ride expires it is not shown in available rides.

**Frequency of Use** High

**Priority p1’**

**Use Case 6: Request custom ride**

**Primary Actor**: Rider

**Scope:**  ShareNRide web based System

**Level:**  User level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** Driver logged into the system

**Minimal guarantees:** Rider Request a costumn ride as per his requirement..

**Success guarantees:** Posted message is displayed .

**Trigger:** No suitable ride available. Rider wants to a ride.

**Main success scenario**:

1. Rider login into the system.
2. No suitable available rides found.
3. Rider enters the ride details and posts.
4. Ride is inserted into the table.
5. Request rides are seen by the drivers..

**Extension**: **1.**  If valid ride details are not provided error message is displayed.

2. If the date of the ride expires it is not shown in available rides.

**Frequency of Use** High

**Priority p1**

**Use Case 7: Join Ride**

**Primary Actor**: Rider, Driver

**Scope:**  ShareNRide web based System

**Level:**  User level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** Ride available for riders with vacant seats.

**Minimal guarantees:** Ride available to the user.

**Success guarantees:** Ride is added to current rides of the user.

**Trigger:** Availability of rides.

**Main success scenario**:

1. User login into the system.
2. Looks for suitable ride.
3. Rider joins the ride and gets added to the current rides.

**Extension:** 1. If no seats are available user can request for new rides.

**Frequency of Use High**

**Priority p1.**

**Use Case 8: Use coupons**

**Primary Actor**: Driver, Rider

**Scope:**  ShareNRide web based System

**Level:**  user level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** Must be a Logged in to the system and has bought coupons.

**Minimal guarantees:** User able to avail coupon facility

**Success guarantees:** User can buy or apply coupon to ride.

**Trigger:** Availability of rides and coupons.

**Main success scenario**:

1. User logins into the system.
2. User sees available rides .
3. User Has coupons available in his account.
4. Coupon is applied to the ride.
5. Ride is added to the user current rides.

**Extension: 1.** If no coupon is available user is asked to buy coupons

2. If no coupons available discount is not applied

**Frequency of Use**  Moderate

**Priority** p2

**Use Case 9: Accept Ride**

**Primary Actor**: Driver

**Scope:**  ShareNRide web based System

**Level:**  User level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** Scheduled rides .

**Minimal guarantees:** Ride gets added too users rider and driver

**Success guarantees:** Availability of ride to the posted rider.

**Trigger:** Driver searches for the requested custom rides.

**Main success scenario**:

1. Driver logs into the system.
2. Driver search for the posts on customized rides.
3. Driver enters the price based on the distance and ride gets added to rider and drivers’ current rides.
4. Success message is displayed once accepted.

**Extensions: 1.** If Driver has another ride scheduled within 2 hours interval driver cannot accept this ride.

**Frequency of Use High**

**Priority**  p2

**Use Case 10: Start/stop ride**

Status of the driver and ride when starts a ride.

**Primary Actor**: Driver

**Scope:**  ShareNRide web based System

**Level:**  user level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** 1. User logged in the system and joined a ride.

2.user has a ride scheduled in current rides.

**Minimal guarantees:** Ride gets inserted in Both driver/rider history.

**Success guarantees:** Once ride starts the status of ride gets updated and when ends it gets updated in driver/rider profiles.

**Trigger:** Ride accepted by the user.

**Main success scenario**:

1. Driver logged into the system.
2. Driver starts and end the ride.
3. Ride is removed from the available rides list.

**Frequency of Use High**

**Priority**  P3

**Use Case 11: Mange driver car**

**Primary Actor**: Driver

**Scope:**  ShareNRide web based System

**Level:**  user level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** Must have taken rides

**Minimal guarantees:** User add his car .

**Success guarantees:** Driver uses this car to accept/post rides.

**Trigger:** Need to accept/post a ride.

**Main success scenario**:

1. User logins into the system.
2. User must be a driver.
3. Driver enters the car details.
4. If already entered driver can edit car details.
5. Car details are added to the ride when driver post/accept rides.

Extensions: If driver has no car details in the system he cannot join a ride.

Driver must foolow the validations to insert/update the car.

**Frequency of Use Moderate**

**Priority** p2

**Use Case 12: view Ride history**

**Primary Actor**: Driver, Rider

**Scope:**  ShareNRide web based System

**Level:**  user level

**Stakeholder’s and interests**: ShareNride app

**Precondition:** Driver finished the rides.

**Minimal guarantees:** Queries rides database using user id.

**Success guarantees:** User able to see his/her rides history.

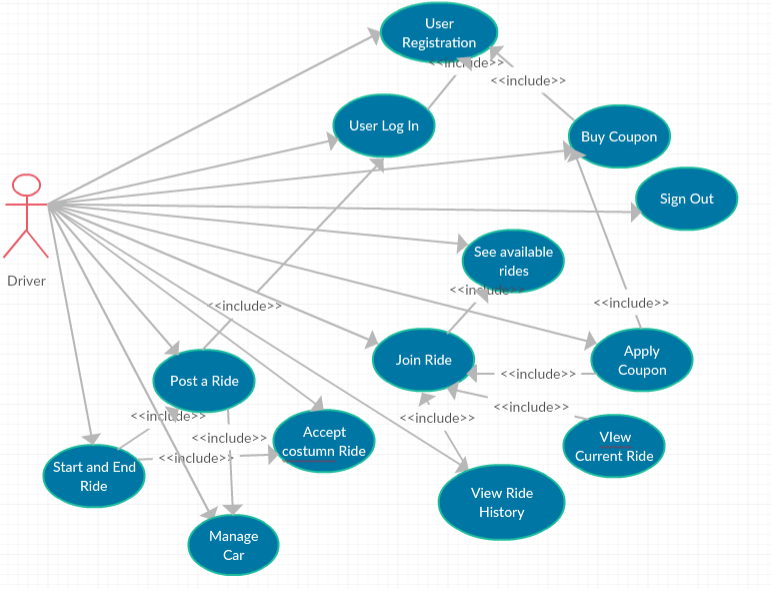
**Trigger:** User Navigates to my Account.

**Main success scenario**:

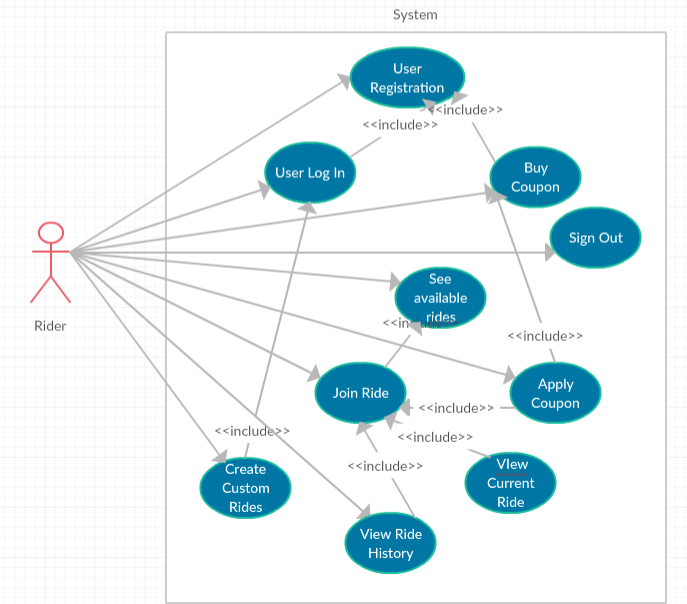
1. User logins into the system.
2. User has taken rides earlier.
3. Display rides queried form the database.

**Frequency of Use Moderate**

**Priority** p2



**Driver Use cases**



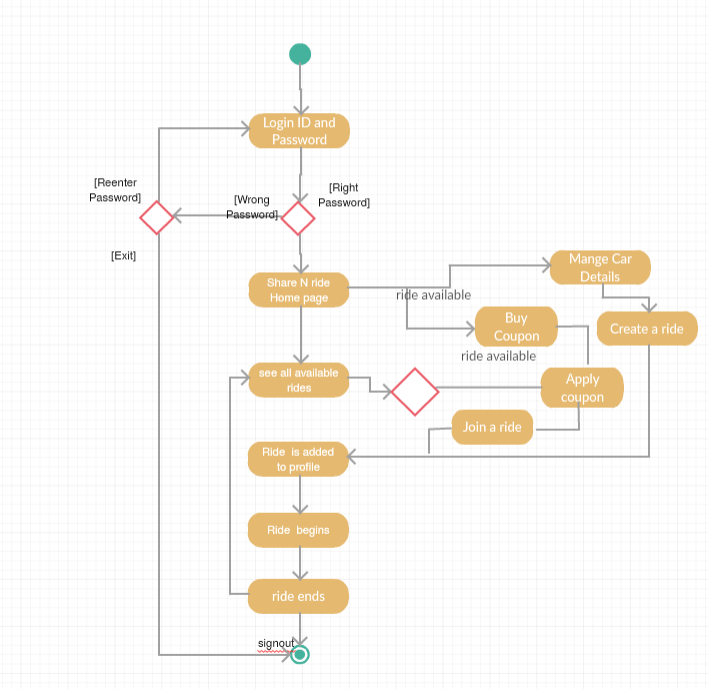
**Rider Use cases:**

**Activity Diagrams:**

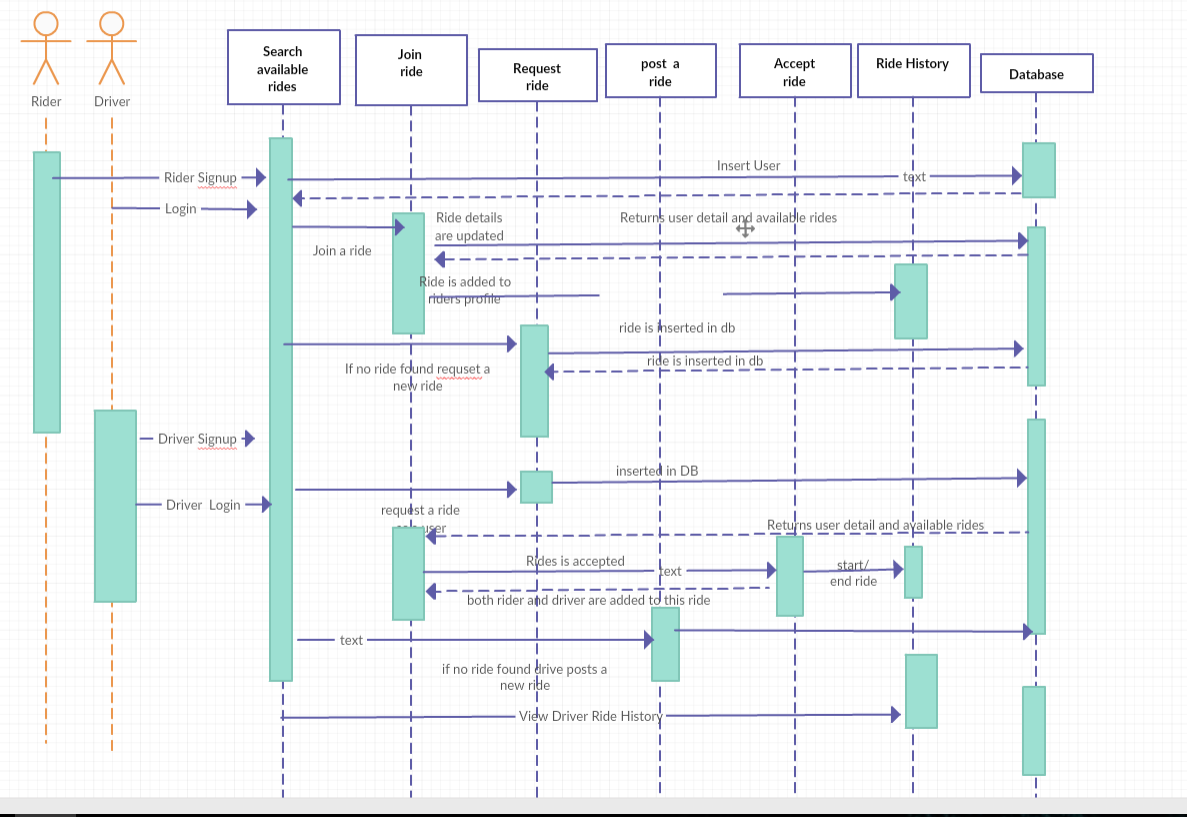


Rider Activity diagram

**Driver Activity diagram:**



**Sequence Diagrams:**



**ER Diagram:**



**Help Log:**

Angular 4: courses from YouTube, Udemy

Bootstrap : youtube,W3schools ,

Php: PDO, Udemy course ,

**Inperson Help:**

Naresh Krishnamaneni— Cousin, Dallas, Texas,

Consulted and learned angular 2 applications developments and how to set up the backend php for the project. Usage of postman to test api services.

Raghu- Brother ,irving, texas

Helped me with bootstrap and angular 4 applications has been a great help with routing in angular 4 and with dependency injections and resolving issues from time to time.

**Initial setup issues:**

First: app routing Module setup

Had issues with the placement or import of app-routing.module.ts .Problem with loading home component .the pathMatch attribute concept.

Faced issue with navigation bar input and communiation between the components.

Links and references:

template driven forms:

<https://codecraft.tv/courses/angular/forms/template-driven/><https://angular.io/guide/dependency-injection>

**SQL Queries used in PHP apis:**

**Registering the User:**

$query = "INSERT INTO " . $this->table\_name . " SET userType=:userType,firstname=:firstname, lastname=:lastname,email=:email,password=:password,mobile=:mobile";

**Login:** $query = "SELECT \* FROM " . $this->table\_name . " WHERE email = ? and password= ? LIMIT 0,1";

**Ride post:**

$query = "INSERT INTO ". $this->table\_name." (user\_id,seats,destination,origin,date,mobile,userType,postedBy,price) VALUES ".$this->UserId."','".$this->Seats."','".$this->DestinationID."','".$this->Origin."','".$this->RideDate."','".$this->UserMobile."'.".$this-UserType."','".$this->postedBy."','".$this->price."')";

**Ride Update:**

$query = "UPDATE " . $this->table\_name . " SET seats=seats-1 where ride\_id=".$this->RideID;

**Ride Delete:**

$query="DELETE FROM ".$this->table\_name." WHERE ride\_id=".$this->RideID;

**Get Available Rides:**

$query = "SELECT \* FROM " . $this->table\_name . " WHERE ( date>CURRENT\_DATE and seats>0 and UserType='Driver')";

**Coustumnridesavail\_get:**

$query = "SELECT \* FROM " . $this->table\_name . " WHERE ( date>CURRENT\_DATE and seats>0 and UserType='Rider')";

**costumnRide\_post:**

$query = "INSERT INTO ". $this->table\_name ." (user\_id,seats,destination,origin,date,mobile,userType,postedBy,price) VALUES

('".$this->UserId."','".$this->Seats."','".$this->DestinationID."','".$this->Origin."','".$this->RideDate."','".$this->UserMobile."', '".$this->UserType."','".$this->postedBy."','0.00')";

**getuserCoupon():**

$query = "SELECT \* FROM " . $this->table\_user\_coupons." where coupon\_rides>0 and user\_id=".$this->UserId;

**buyCoupon():**

$query="INSERT INTO " .$this->table\_user\_coupons."(user\_id,coupon\_code,coupon\_type,coupon\_validity,coupon\_rides,coupon\_price) VALUES ('".$this->UserId."','".$this->coupon\_code."','".$this->coupon\_type."','" .$this->coupon\_validity."','".$this->coupon\_rides."','".$this->coupon\_price."')";

**getAllCoupons() :**

$query = "SELECT \* FROM " . $this->table\_name1;

**Currentrides\_post()** $query = "INSERT INTO ". $this->table\_name ." (ride\_driver\_name,ride\_rider\_name,ride\_user\_id,ride\_driver\_id,ride\_id,origin,destination,ridetime,seats,price) VALUES '".$this->DriverName."','".$this->RiderName."','".$this->RiderId."','"$this->DriverId."','".$this->RideID."','".$this->Origin."','".$this->DestinationID."','".$this->RideDate."','".$this->Seats."','".$this->Price."')";

**currentrides\_update():**

$query = "UPDATE " . $this->table\_name . " SET Status=:Status WHERE RideID=:RideID";

**currentdriverrides\_get():**

$query = "SELECT \* FROM " . $this->table\_name . " WHERE ( ride\_driver\_id='" .$this->DriverId."'and ridetime>=CURRENT\_DATE)";

**currentRiderrides\_get():**

$query = "SELECT \* FROM

" . $this->table\_name . " WHERE ( ride\_user\_id='" .$this->RiderId."'and ridetime>=CURRENT\_DATE)";

**currentrides\_delete():** $query="DELETE FROM ".$this->table\_name." WHERE current\_rideid=".$this->current\_ride\_id;

**Historyrides\_post:** "INSERT INTO ride\_history (ride\_driver\_name,ride\_rider\_name,ride\_user\_id,ride\_driver\_id,ride\_id,origin,destination,ridetime,rideendtime) VALUES

('".$this->DriverName."','".$this->RiderName."','".$this->RiderId."','" $this->DriverId."','".$this->RideID."','".$this->Origin."','".$this->DestinationID."','".$this->RideDate."','".$newTime."')";

**Historydriverrides\_get():**

$query = "SELECT \* FROM ride\_history WHERE ( ride\_driver\_id='" .$this->DriverId."')";

**HistoryRiderrides\_get():**

$query = "SELECT \* FROM ride\_history m WHERE ( ride\_user\_id='" .$this->RiderId."')";

**Validating the driver availability:**

$query="select \* from " . $this->table\_name . " WHERE user\_id='" .$this->UserId. "'and userType='Driver' and date between DATE\_ADD('".$this->RideDate."', INTERVAL -2 HOUR) and DATE\_ADD('".$this->RideDate."', INTERVAL 2 HOUR)";

**Validating user availability to accept rides:**

$query="select \* from " . $this->table\_name . " WHERE ride\_driver\_id='" .$this->DriverId. "'and ridetime between DATE\_ADD('".$this->RideDate."', INTERVAL -2 HOUR) and DATE\_ADD('".$this->RideDate."', INTERVAL 2 HOUR)";

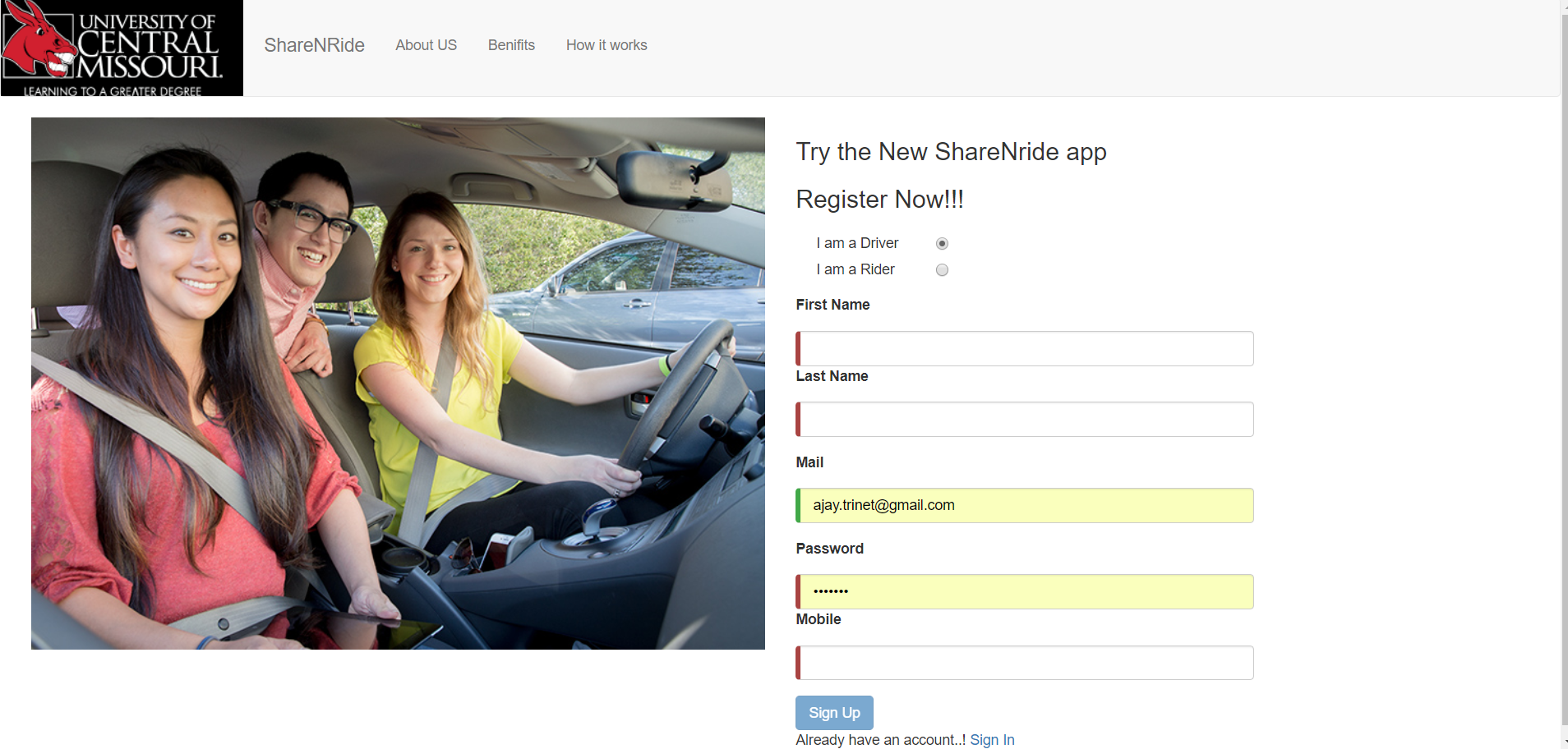
**Initial setup issues:**

First: app routing Module setup

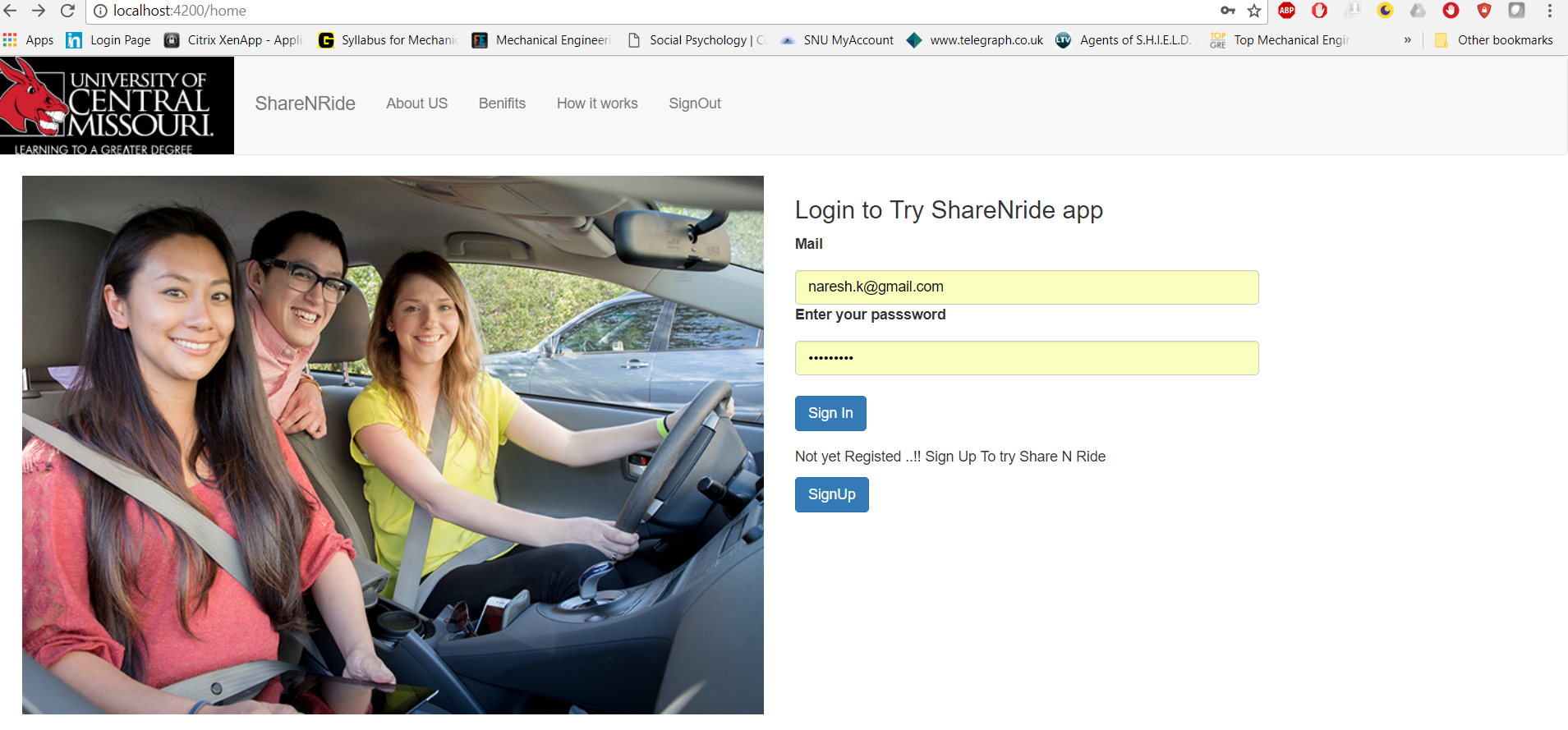
Had issues with the placement or import of app-routing.module.ts .Problem with loading home component .the pathMatch attribute concept.

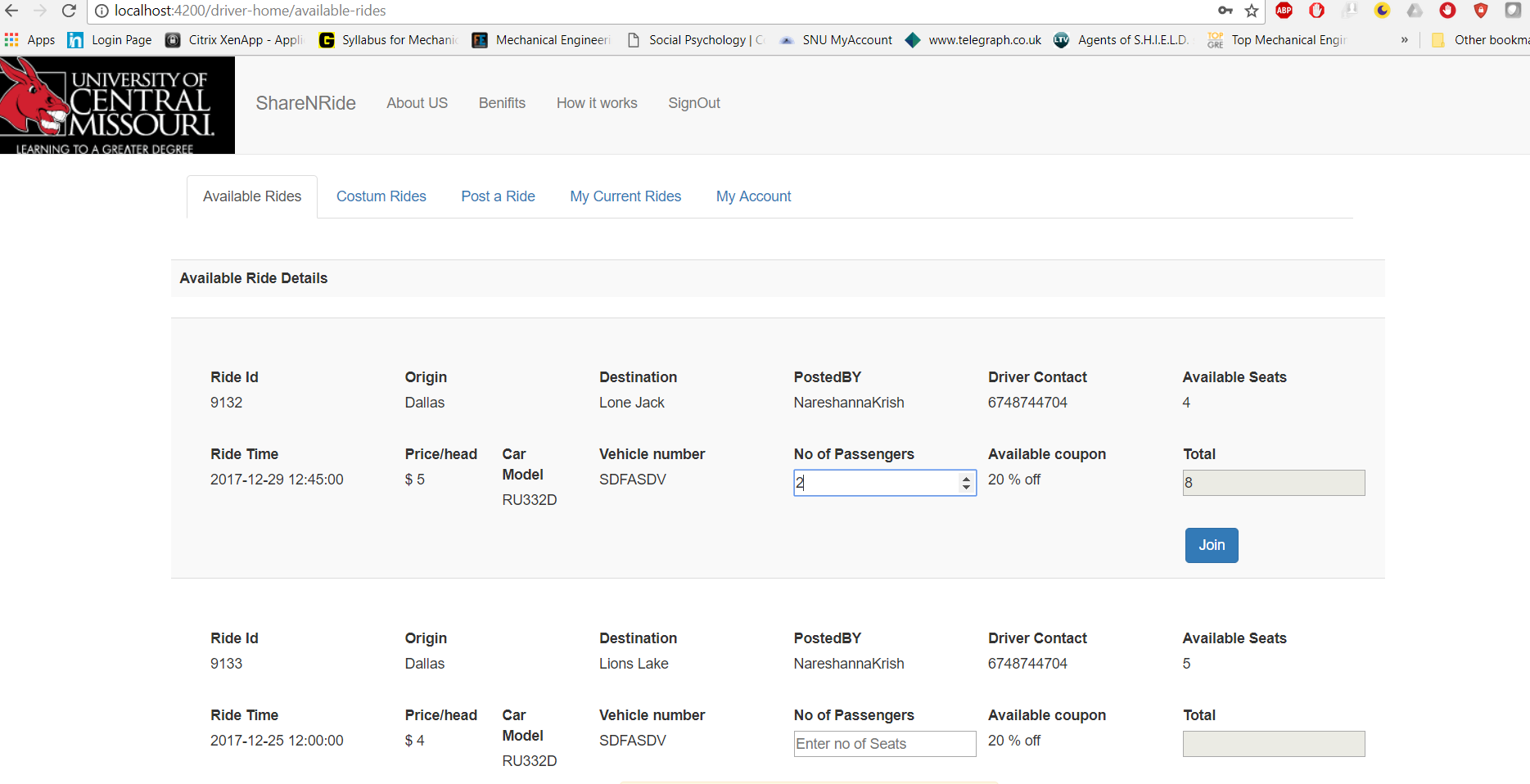
Faced issue with navigation bar input and communiation between the components.

**Project Screen Shots:**

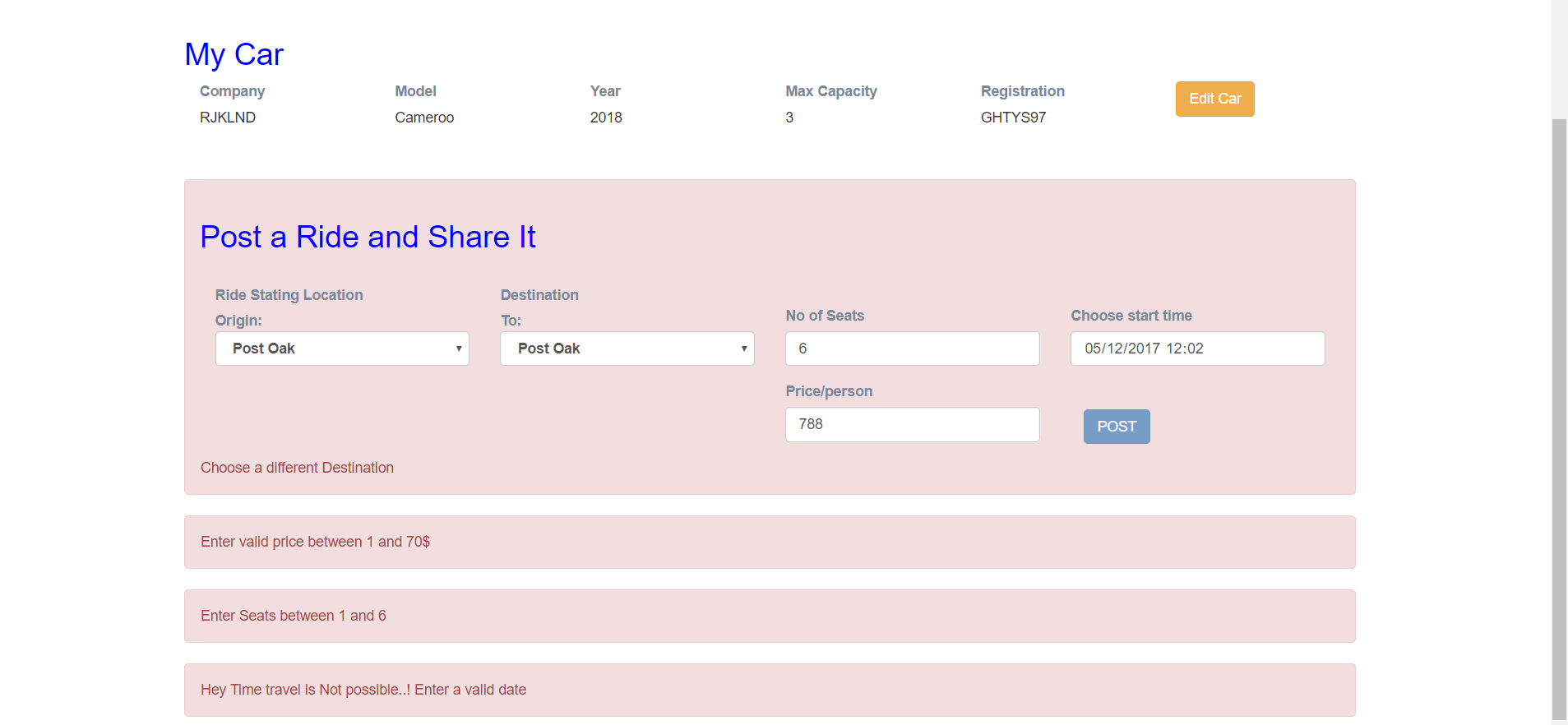


**Login:**

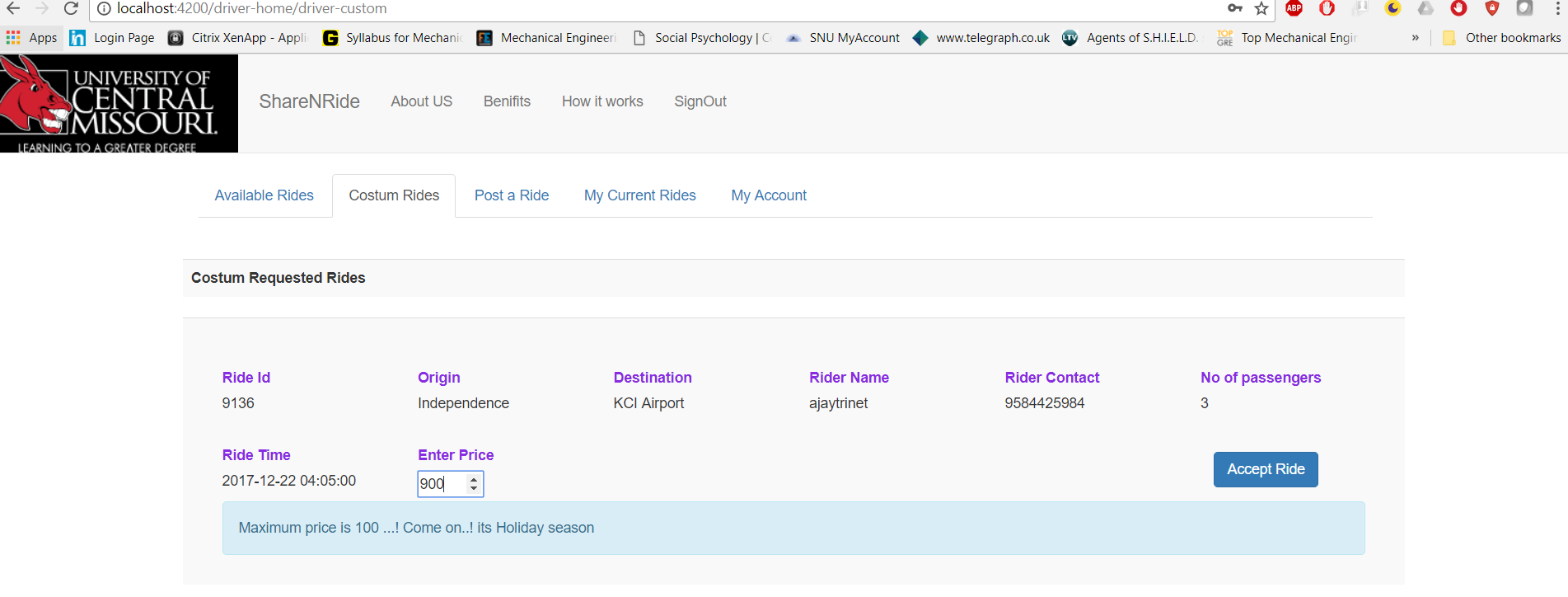


**Available rides and join ride.**

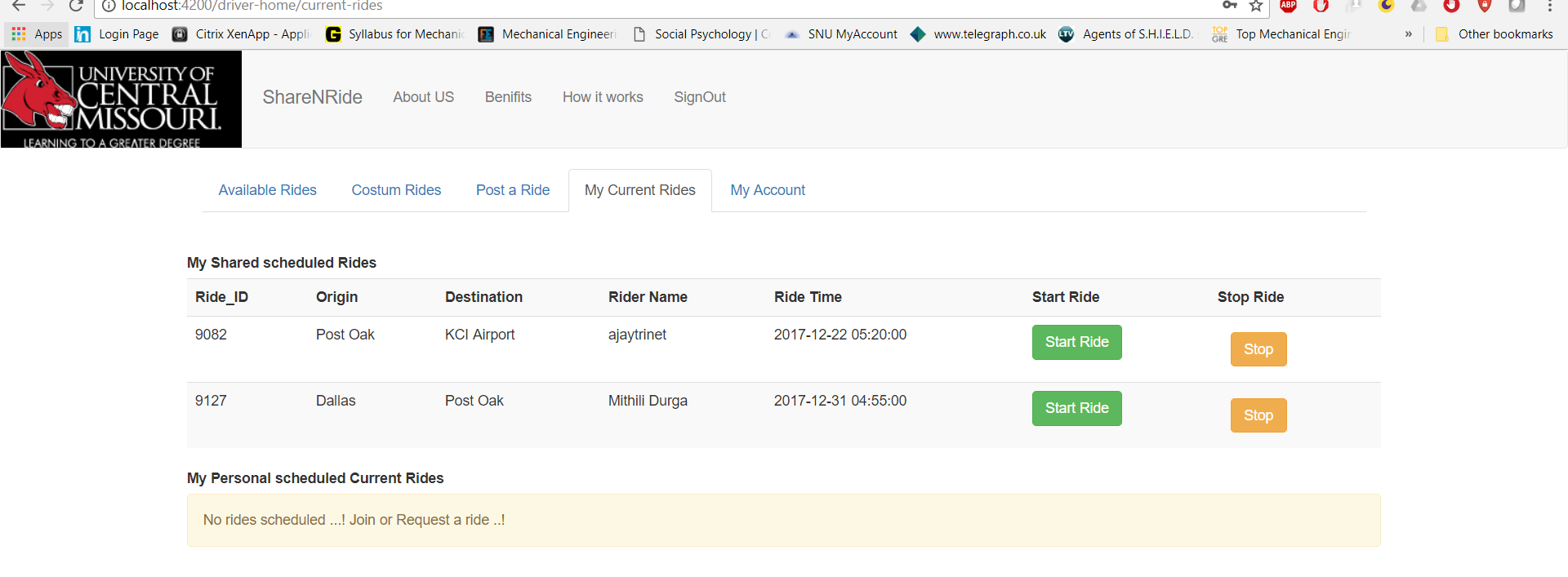
**Post ride and Manage car detsails.**



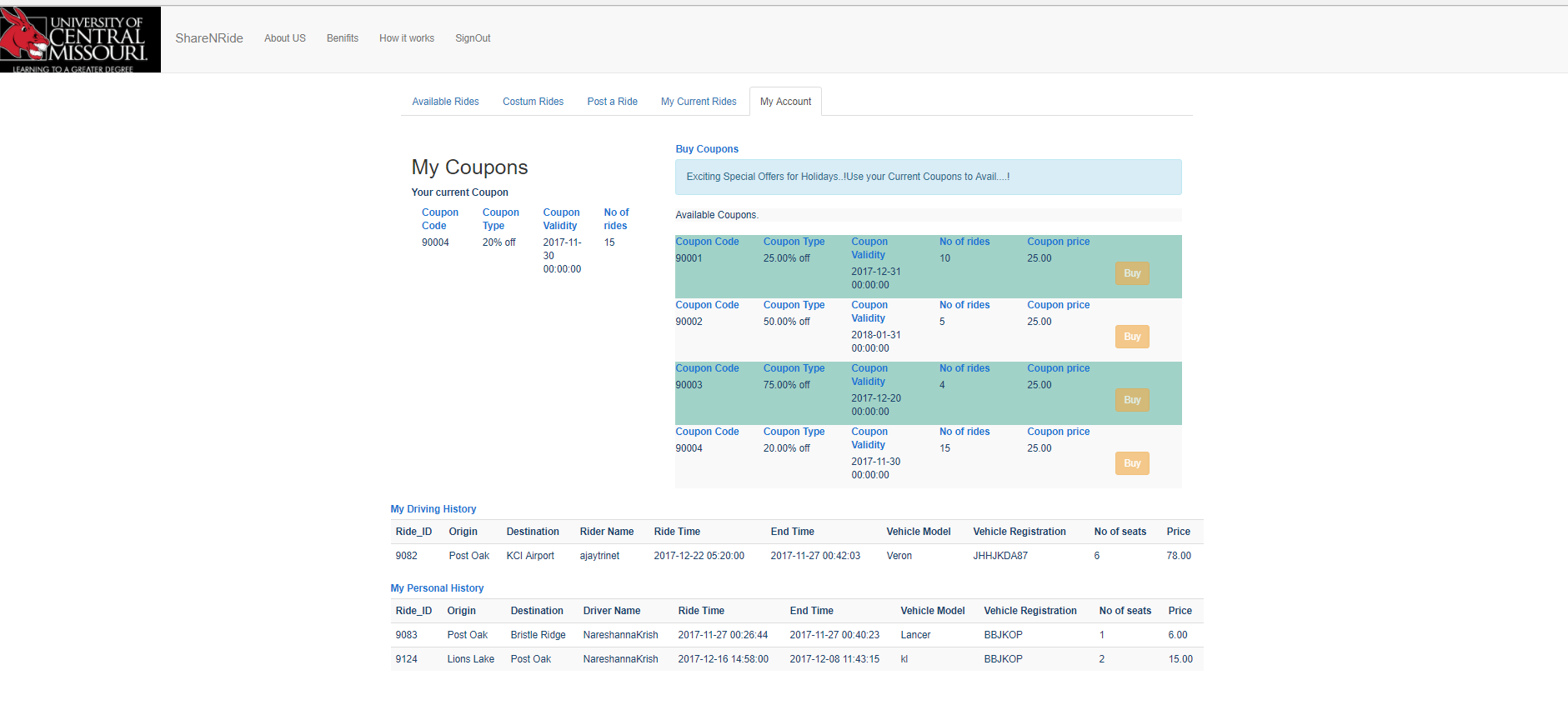
**Accept Riders custom rides:**



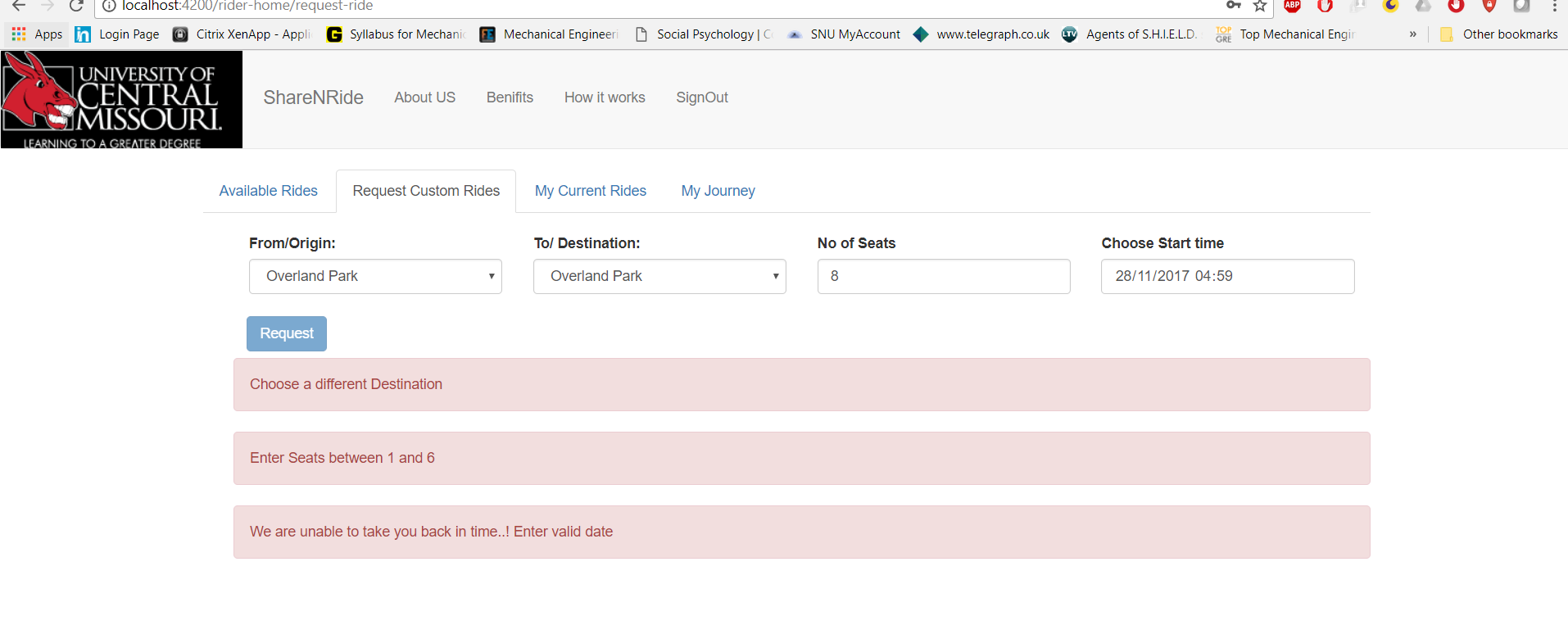
**Start/ end Rides: My current rides**



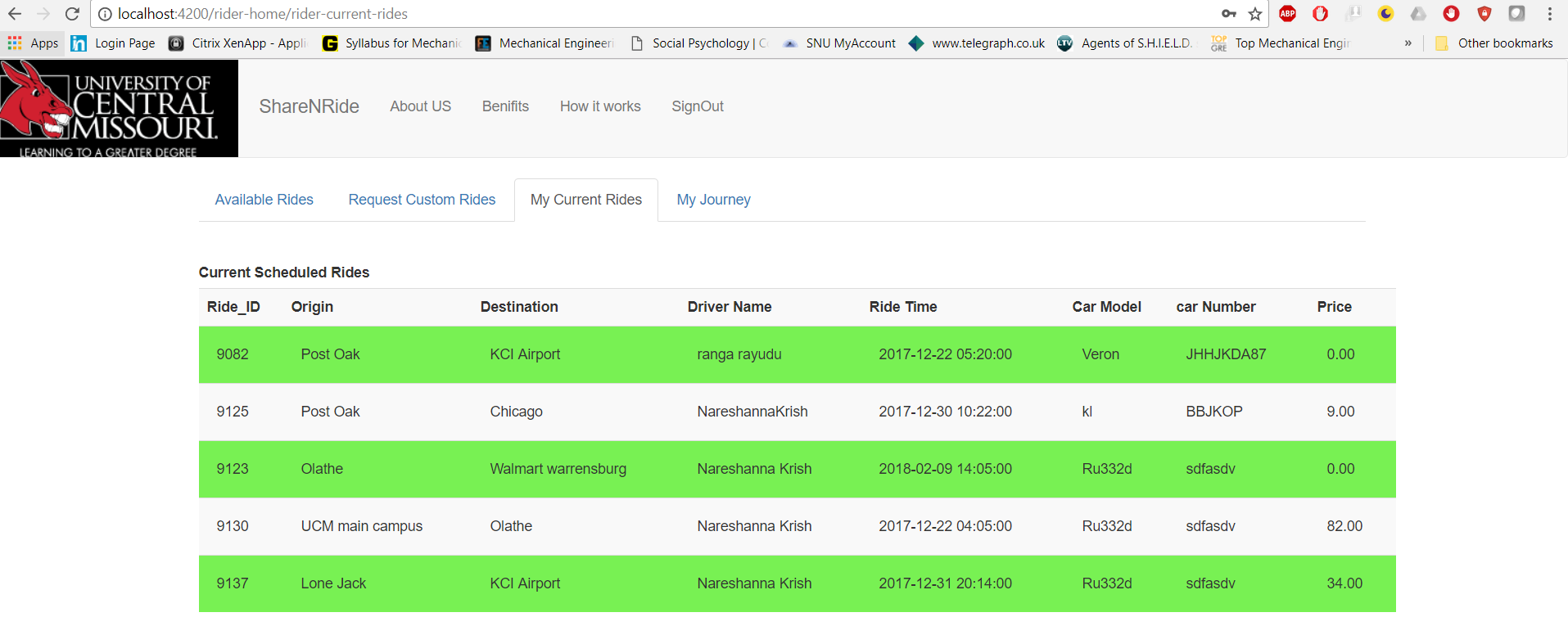
**Driver My account/ history/buy coupons/ mange offers.**



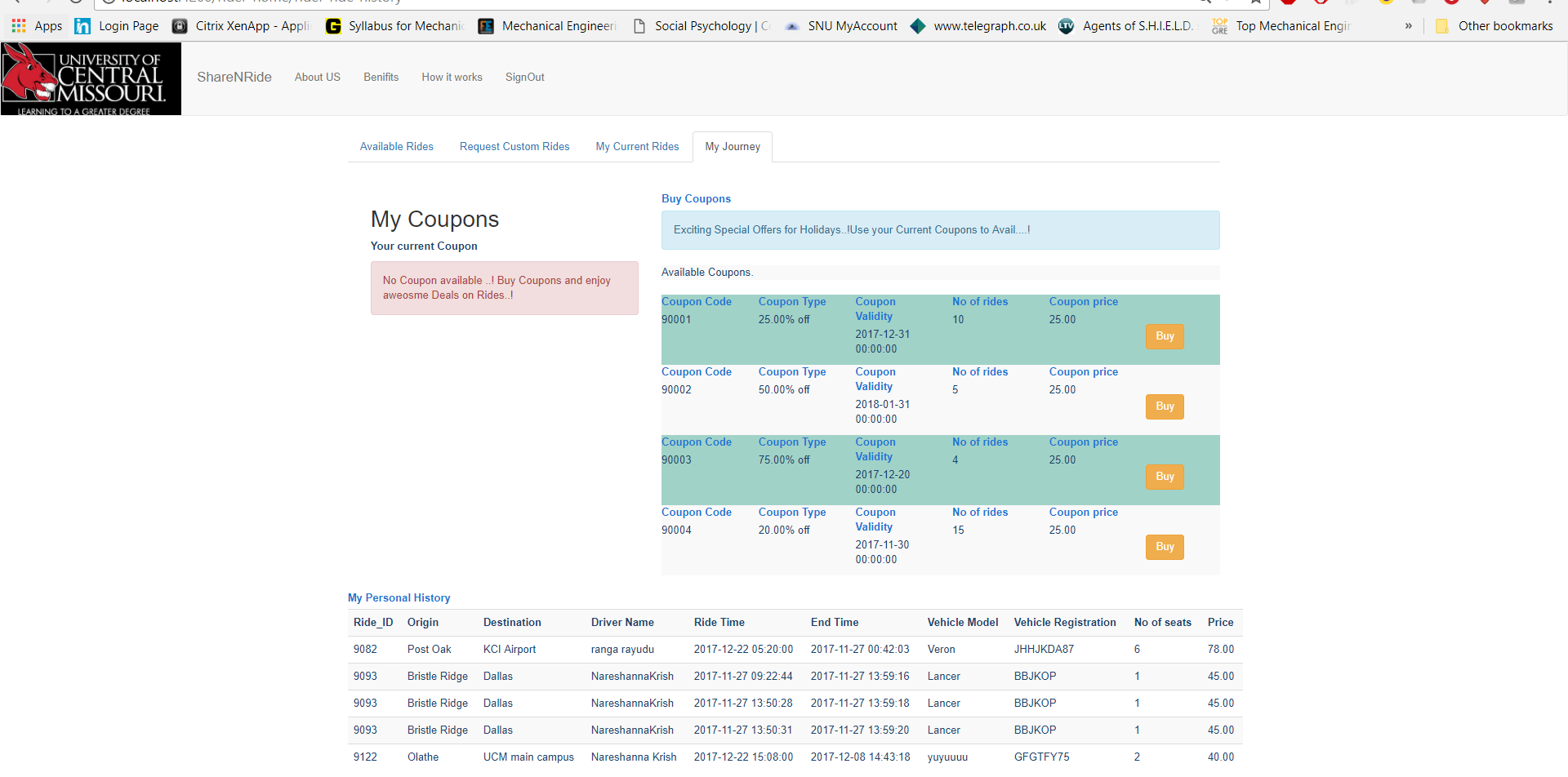
**Rider Costumn ride request:**



**Driver scheduled rides:**



**Rider Historty/account/manage offers**



**Links and references:**

template driven forms:

<https://angular.io/docs>

<https://codecraft.tv/courses/angular/forms/template-driven/><https://angular.io/guide/dependency-injection>

<https://angular.io/guide/component-interaction>

<https://www.youtube.com/watch?v=8bE0H495pu8>

<https://www.youtube.com/watch?v=kFTmoLm9Jwg>

Refered Php MDN.

<http://php.net/manual/en/pdo.connections.php>