

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

CARGO is a web application which is developed using industrial standard technologies like HTML JavaScript, PHP and SQL. It is a web-based aggregator system that helps people to search for suitable ride or a suitable ride partner. It allows user to create their own specific account with a specific id and access it with a password. Once the user is entered into the account he/she can select either the ride or drive option according to their need, with a specific source and destination. This website compares the details of both the users.

1.2 PROBLEM STATEMENT

As time passes, it is sure to have an increased number of vehicles on the roads. We could stabilize the motion by analyzing the recent study report released by Govt. of India. In such a scenario, vehicle pooling services; especially carpooling and bike pooling are the only possible solution to solve the rush on the roads. Pooling Services aren't a new to us. Here, team CarGo is trying to build a new way of approach to it by allowing the rider and driver to meet online beforehand. CarGo also helps in reducing the traffic and thereby helps in building a carbon-free society

1.3 SCOPE AND RELEVANCE

As CarGo connects people who drive the way with those who wish to have a shared trip, it also helps in building healthy community in the society.

1.4 OBJECTIVE

The objective of the project is to build a web application based aggregator service that connects people who drive the way, with those who are interested in having a shared trip; thereby acting as interface between them.

SYSTEM ANALYSIS

CHAPTER 2

SYSTEM ANALYSIS

2.1 INTRODUCTION

System analysis is the process of gathering and interpreting facts and diagnosing problems using the information improvements on the system.

The system analysis is a problem-solving activity that requires intensive communication between the user and the system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute, detailed and analyzed. The system is viewed as a whole and inputs to the system are identified. The outputs from the organization are traced to the various processing that the input phase.

Analysis is the first step towards solving the problem statement. It deals with devising a precise, concise, understandable and correct model of the real system. Analysis helps to understand the requirements and the real world environment in which the system will exist. A detailed study of this process can be made by various techniques like interviews, questionnaires etc. The data collected by these resources must be scrutinized to arrive at a conclusion. The conclusion is an understanding of how the system is subjected to close study and problem areas are identified. The designer functions as a problem solver and tries to sort out the difficulties that enter the fact.

2.2 EXISTING SYSTEM

The present pooling service is like when a person books a taxi, the taxi which he books will come to the place which has been given and will pick the user to the destination which he/she wants. But our online application provides two options RIDE or DRIVE where the user can hire a taxi or can find a suitable travel partner using our application.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- It is high cost (compared to our application the older pooling services were of high cost)
- No information about the driver

2.3 PROPOSED SYSTEM

Taking into account the various drawbacks that inherit in the existing system, a new system is proposed which eliminated them. A computerized system is adopted to resolve these problems.

In the previous pooling services where introduced in the market and most people didn't have an idea about this system but when people came to know about these services, understanding its value and effect in society, many were launched but, all of them followed same features of the previous system without any change.

2.3.1 ADVANTAGE OF PROPOSED SYSTEM

- We can book a taxi using our smartphone from the current destination.
- Instead of calling or waiting a taxi we can use these application to book a taxi from far away location
- User friendly

2.4 FEASIBILITY STUDY

In any project, feasibility analysis is a very important stage: here the project is checked for the feasibility. Any project may face scarcity in resources, time for workforce. Hence all these are to be studied in detail and a conclusion should be drawn whether the project under consideration is feasible or not. This analysis is a test of the proposed project, regarding its workability, impact on users and clients and resource management. Feasibility and risk involved are inversely related to each other. The main objective of the feasibility is to test the technical, social and economic feasibility of a project.

2.4.1 TECHNICAL FEASIBILITY

It measures the feasibility of the particular technical solution and the availability of the technical resources and the expertise. This is concerned with the specifying equipment and software satisfy the user requirement, the technical needs of the system may vary considerably, but might include. The facility to produce output in a given time

To develop this website a computer with a window, internet is needed and technologies like HTML, CSS, SQL, used a front end .PHP and SQL is used as the backend is required.

2.4.2 ECONOMIC FEASIBILITY

Economic feasibility is to know whether expected cost savings, increased revenue, increased profits and reduction in required investment exceed the cost of developing and operating a proposed system. Though order processing system the organization can reduce raw materials cost because this system gives accurate requirements. So, this system is economically feasibility.

2.4.3 Operational Feasibility:

Proposed projects are beneficial only if they can be turned into information system that will meet the operating requirement of the organization. The test of feasibility asks if the system will work when it is developed and installed. Some of the important question that are useful to test the operational feasibility of a project are given below.

- ✓ Is there sufficient support for the project from the management? From user? If the present system well liked and used to the extent that people would not be able to see reasons for a change, there may be a resistance.
- ✓ Issues that appear to be quite minor at the early stage can grow into major problem after implementation.

2.5 SOFTWARE ENGINEERING PARADIGM APPLIED

This establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines is called software engineering.

Software engineering is the discipline whose aim is:

1. Production of quality software.
2. Software that is delivered on time.
3. Cost within the budget.
4. Satisfies all requirement

Software process is the way in which we produce the software. Apart from hiring smart, knowledgeable engineers and buying the latest development tools, effective software development process is also needed, so that engineers can systematically use the best technical and managerial practices to successfully complete their projects.

A **software life cycle** is the series of identifiable stages that a software product undergoes during its lifetime. A software lifecycle model is a descriptive and diagrammatic representation of the software life cycle. A life cycle model represents all the activities required to make a software product transit to life cycle phases. It also captures the order in which these activities are to be taken.

Life Cycle Models

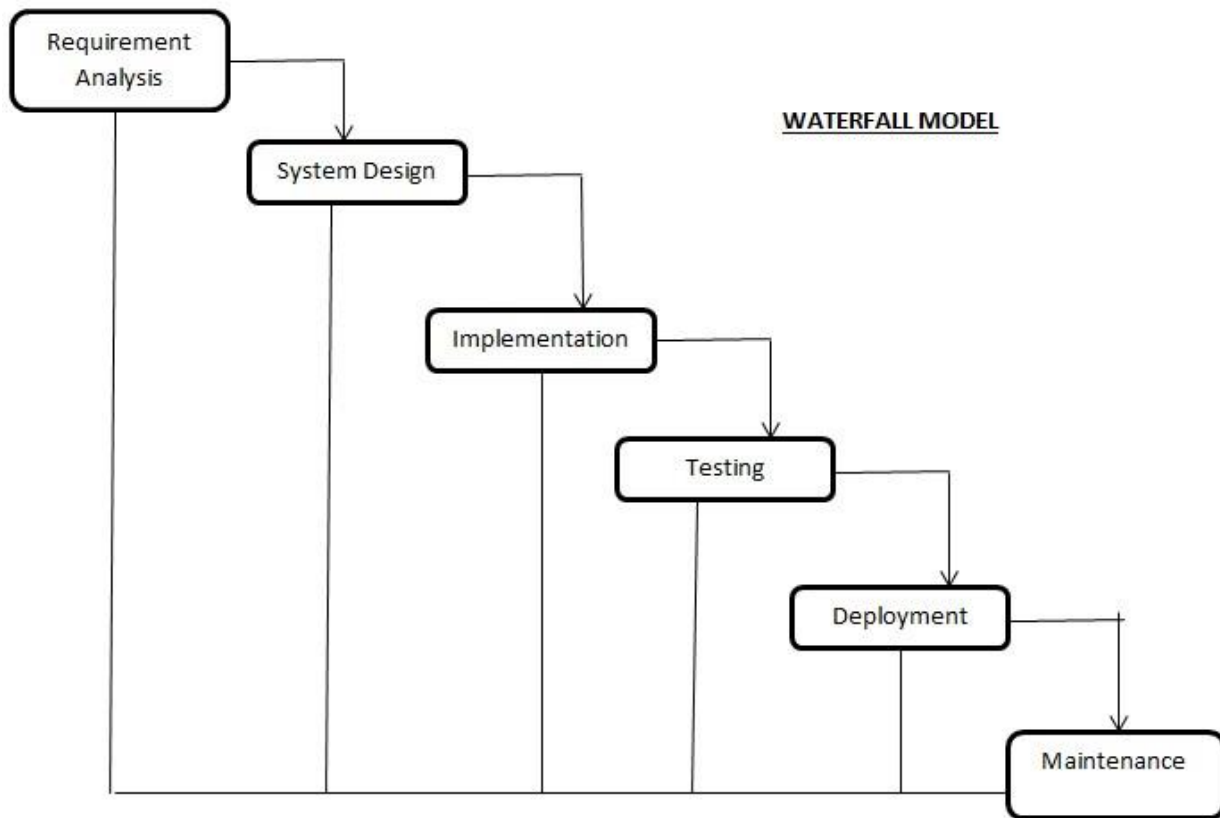
There are various Life Cycle Models to improve the software processes

1. Waterfall Model
2. Prototype Model
3. Iterative Enhancement Model
4. Evolutionary Model
5. Spiral Model

In this project ***waterfall model*** is followed.

Waterfall approach was first SDLC Model to be used widely in software engineering to ensure success of the project. In “The Waterfall” approach, the whole process of software development is divided into separate phrases. In this Waterfall model, typically, the outcome of one phrase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phrases of the Waterfall Model



The sequential phases in Waterfall model are

- **Requirement Gathering and analysis**- All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification.
- **System Design** – The requirement specification from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

- Maintenance – There are some issues which comes up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name “Waterfall Model”. In this model, phases do not overlap.

SYSTEM DESIGN

CHAPTER 3

SYSTEM DESIGN

3.1 INTRODUCTION

System design is the solution to the creation of a new system. This is the important aspect made up of several steps. System design is the process of developing specifications for a candidate system that meet the criteria established in the system analysis. Major steps in system design is the preparation of the input forms and the output reports in a form applicable to the user.

The main objective of the system design is to use the package easily by a computer operator. System design is the creative act of invention, developing new inputs, a database, off-lines, methods, procedures and output for processing business to meet an organization objective. System design built information gathered during the system analysis.

The complete, efficient and successful system should provide the following in succession:

- ✓ From where should we start
- ✓ Where we have to go
- ✓ Where should we stop

If the project is to be successful we need to answer these questions. The answer of these questions in scheme manner is known as system design. A systematic manner will be followed so as to achieve beneficial result at the end. It involves starting with a vague idea and ultimately developing it up into a useful system. The design phase is transition from a user oriented to a document oriented to the programmers. Software report can be broken in to a series of steps starting with the basic ideas and ending with finished projects.

3.2 DATABASE DESIGN

Database design is the process of producing a detailed data model of a database. The logical model contains all the needed logical and physical design and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system.

NORMALIZATION

The process of normalization is concerned with the transformation of the conceptual schema to a computer represent able form. Normalization reduces the redundancies and anomalies.

THE FIRST NORMAL FORM

First normal form does not allow multi valued and composite valued attributes. It states that the domain of an attribute must include only atomic values and that value of any attribute in a table must be single value form the domain of that attribute.

THE SECOND NORMAL FORM

In second normal form, for relation where primary key contains multiple attributes, on key attribute should not be functionally dependent on a part of the primary key.

THE THIRD NORMAL FORM

In Third normal form, relation should not have a non-key attribute functionally determined by non-key attribute. That is there should be no transitive dependency of a non-key attribute on the primary key.

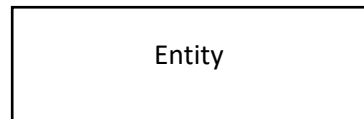
3.21 ENTITY RELATIONSHIP MODEL

Database designs also include ER (Entity relationship model) diagrams. An ER diagram is a diagram that helps to design database in an efficient way. Attributes in ER diagram are usually modeled as an oval with the name of the attribute, linked to the entity or relationship that contains the attribute. Within the relational model the final step can generally be broken down into 2 further steps that of determining the grouping oif information within the system , generally determining

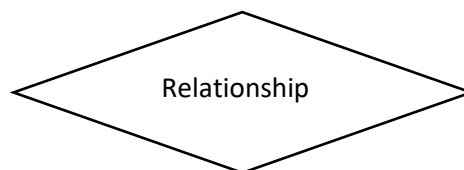
what are the basis objects about which information, is being stored , and then determining the relationships , or objects .This step nis not necessary with an object database .

There are 5 main components of an ERD:

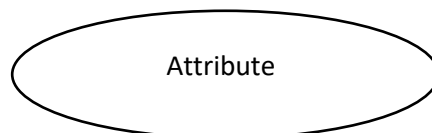
- Entities, which are represented by rectangles. An entity is an object or concept about which you want to store information. A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



- Relationships, which are represented by a diamond shapes, show how to entities share information in the database. In some cases entities can be self-linked .For example, employees can supervise other employees.



- Attributes, which are re[presented by ovals .A key attribute is the unique , distinguishing characteristic of the entity .For example , employee's social security number might be the employee's key attribute .



- Connecting lines, solid lines that connect attributes to show the relationships of entities in the diagram.

TABLE DESIGN**TABLE – 1: REGISTRATION**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION
USER_ID	INT	10	USER IDENTIFICATION
FIRST_NAME	VARCHAR	25	FIRST NAME
LAST_NAME	VARCHAR	25	LAST NAME
EMAIL	VARCHAR	50	EMAIL ID
MOBILE	VARCHAR	10	MOBILE NUMBER
USERNAME	VARCHAR	50	USERNAME
PASSWRD	VARCHAR	50	PASSWORD
MOBILE_VERIFIED	TINYINT	1	STATUS COUNTER

TABLE – 2: TRIP

FIELD NAME	DATATYPE	SIZE	DESCRIPTION
TRIP_ID	INT	10	TRIP IDENTIFICATION
USER_ID	INT	10	USER IDENTIFICATION
SOURCE	VARCHAR	50	TRIP SOURCE
DESTINATION	VARCHAR	50	TRIP DESTINATION
DATEE	DATE		DATE
TIMEE	TIME		TIME
VEHICLE_ID	INT	10	VEHICLE IDENTIFICATION

TABLE – 3: **RIDE**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION
RIDE_ID	INT	10	RIDE IDENTIFICATION
USER_ID	INT	10	USER IDENTIFICATION
SOURCE	VARCHAR	50	TRIP SOURCE
DESTINATION	VARCHAR	50	TRIP DESTINATION
DATEE	DATE		DATE
TIMEE	TIME		TIME

TABLE – 4: **VEHICLE**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION
VEHICLE_ID	INT	11	VEHICLE IDENTIFICATION
VEHICLE_NAME	VARCHAR	50	VEHICLE NAME
PASSENGERS_NO	INT	11	NUMBER OF PASSENGERS
STATUS	TINYINT	1	STATUS COUNTER

TABLE – 5: **PASSENGER**

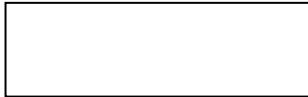
FIELD NAME	DATATYPE	SIZE	DESCRIPTION
P_ID	INT	11	PASSENGER TABLE ID
TRIP_ID	INT	10	TRIP IDENTIFICATION
RIDE_ID	INT	10	RIDE IDENTIFICATION
STATUS	TINYINT	1	STATUS COUNTER

3.3 DATA FLOW DIAGRAMS

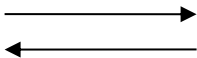
A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modelling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also for visualization of data processing. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

DFD SYMBOLS

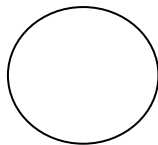
1. A RECTANGLE defines a source or destination of system data.



2. An **ARROW** identifies data flow or data in motion. It is a pipeline through which information flows.



3. A CIRCLE or a BUBBLE represents a process transforms incoming data flow into outgoing data flow.



4. An OPEN RECTANGLE is a data store or at rest or a temporary test repository of data.



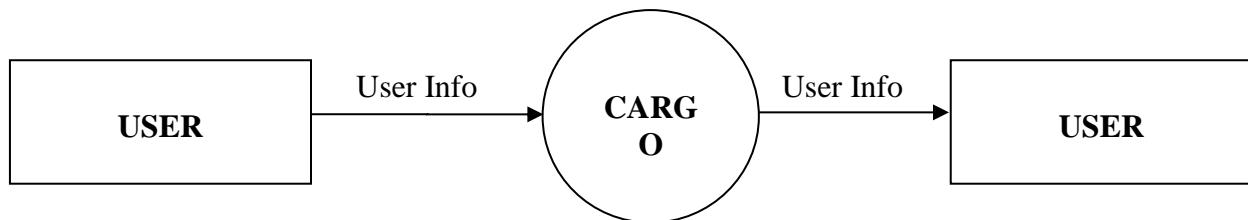
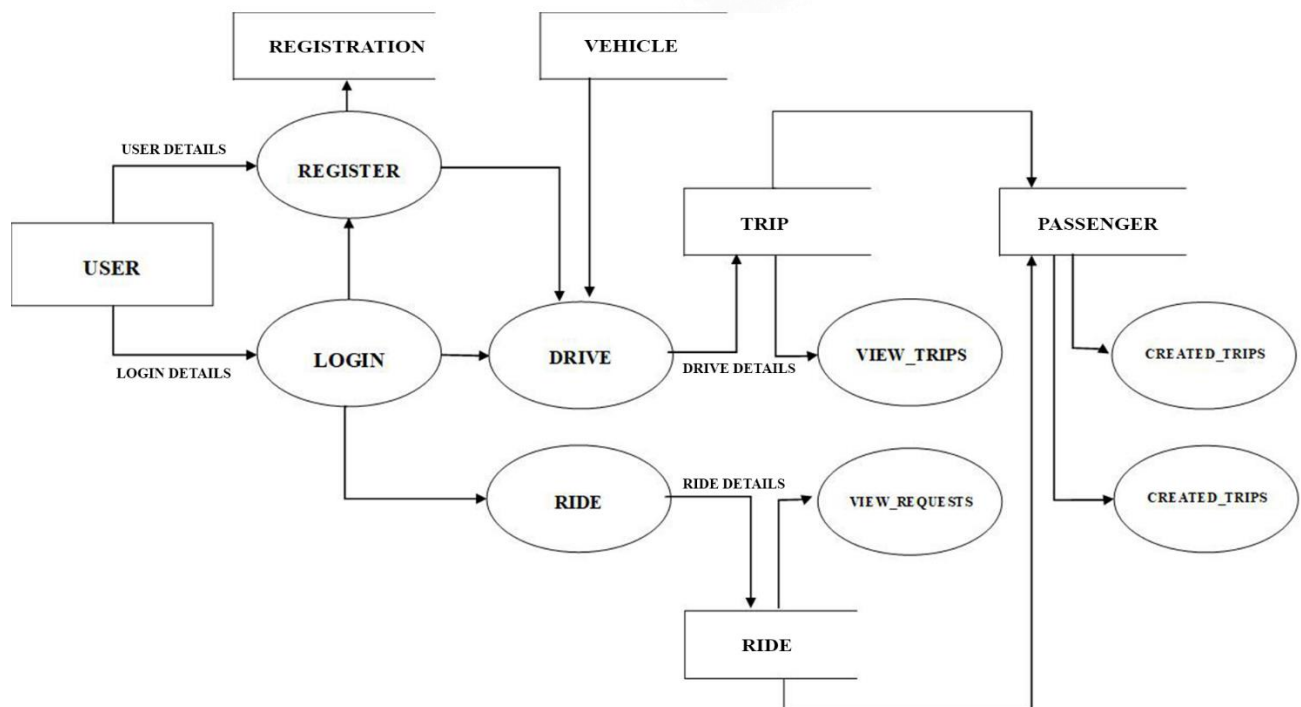
Note that a DFD describes what data flow rather than they are processed, so it does not depend on hardware, software and data structure or file organization by a system.

One of the tools of structured analysis is the diagram. A data flow diagram is a graphical representation of the system. The analyst can use data flow diagram to explain this understanding about the system.

1. Data flows are an initiative way of showing how data is processed by a system.
2. Data flow models are used to show how data flows through a sequence of processing steps.

Four steps are commonly used to construct a DFD.

1. Process should be named and numbered for easy reference.
2. The direction of flow is from top to bottom and from left to right.
3. When a process is exploded into lower level details they are numbered.
4. The names of data stores, sources and destinations are written in capital letters.

LEVEL - 0 DFD**LEVEL - 1 DFD (USER)**

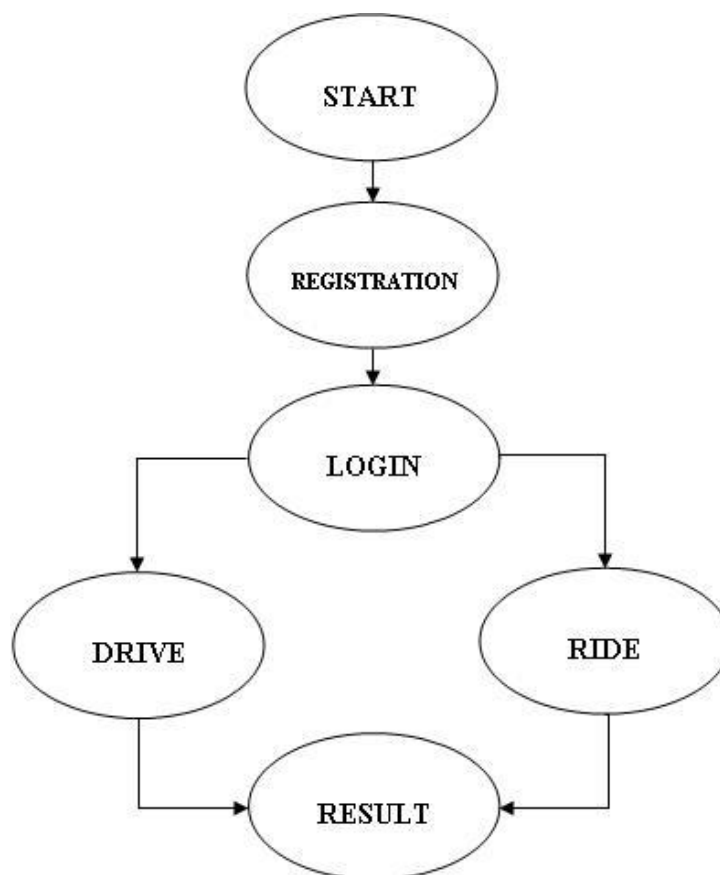
3.4 OBJECT ORIENTED DESIGN-UML DIAGRAMS

UML stands for Unified Modelling Language which is used in object oriented software engineering. Although typically used in software engineering it is a rich language that can be used to model an application structures, behavior and even business processes. There are 14 UML diagram types to help you model this behavior.

3.4.1 Activity Diagram

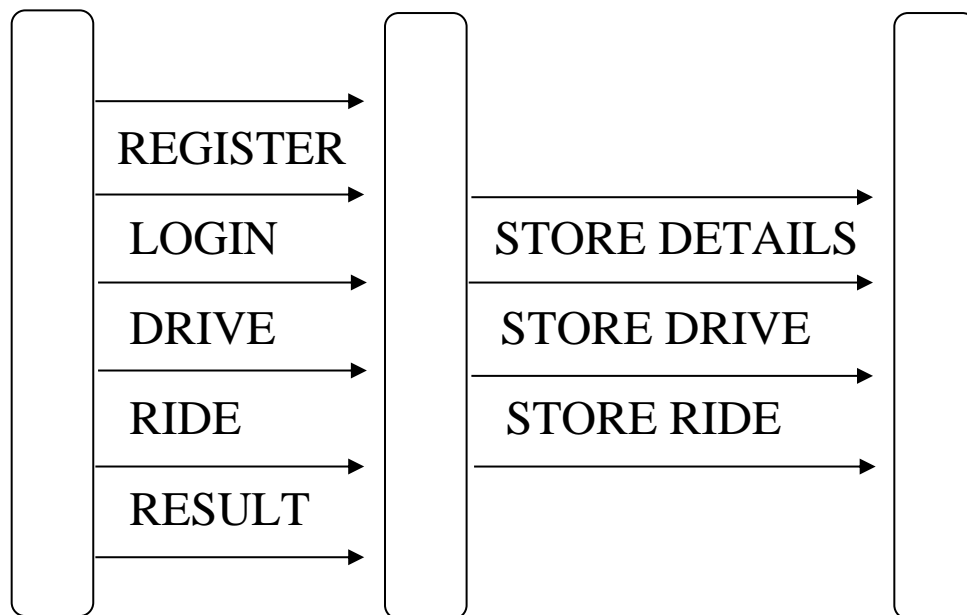
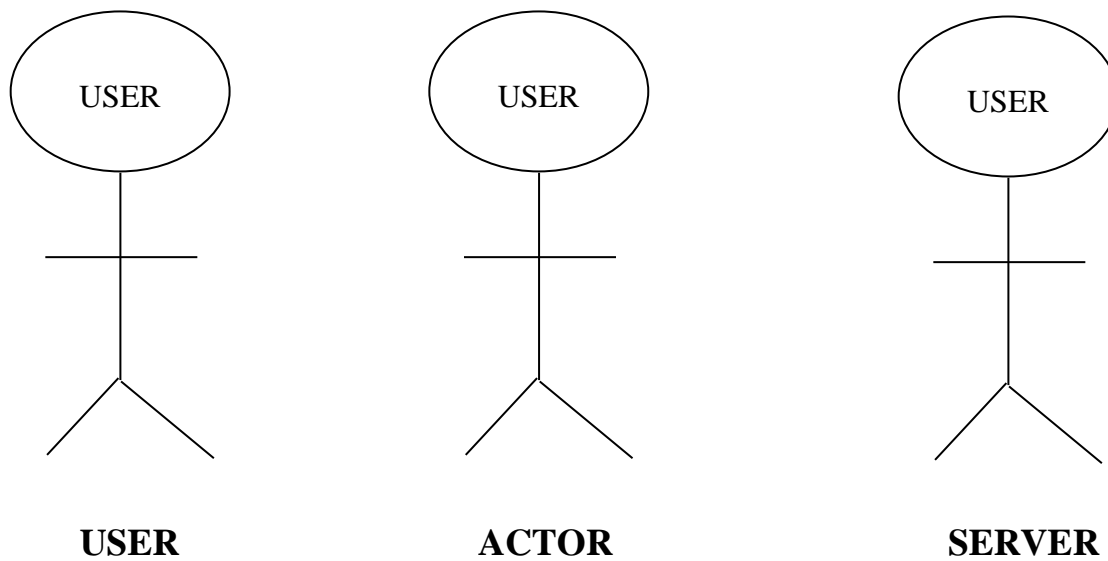
Activity diagrams represent workflows in a graphical way. They can be used to describe business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State Machine Diagrams.

Fig:



3.42 SEQUENCE DIAGRAM

Sequence diagram in UML show object interact with each other and the order those interaction occurs. It's important to note that they show the interaction for a particular scenario .The processes are represented vertically and the interactions are show as arrows .



3.43 Use Case diagram

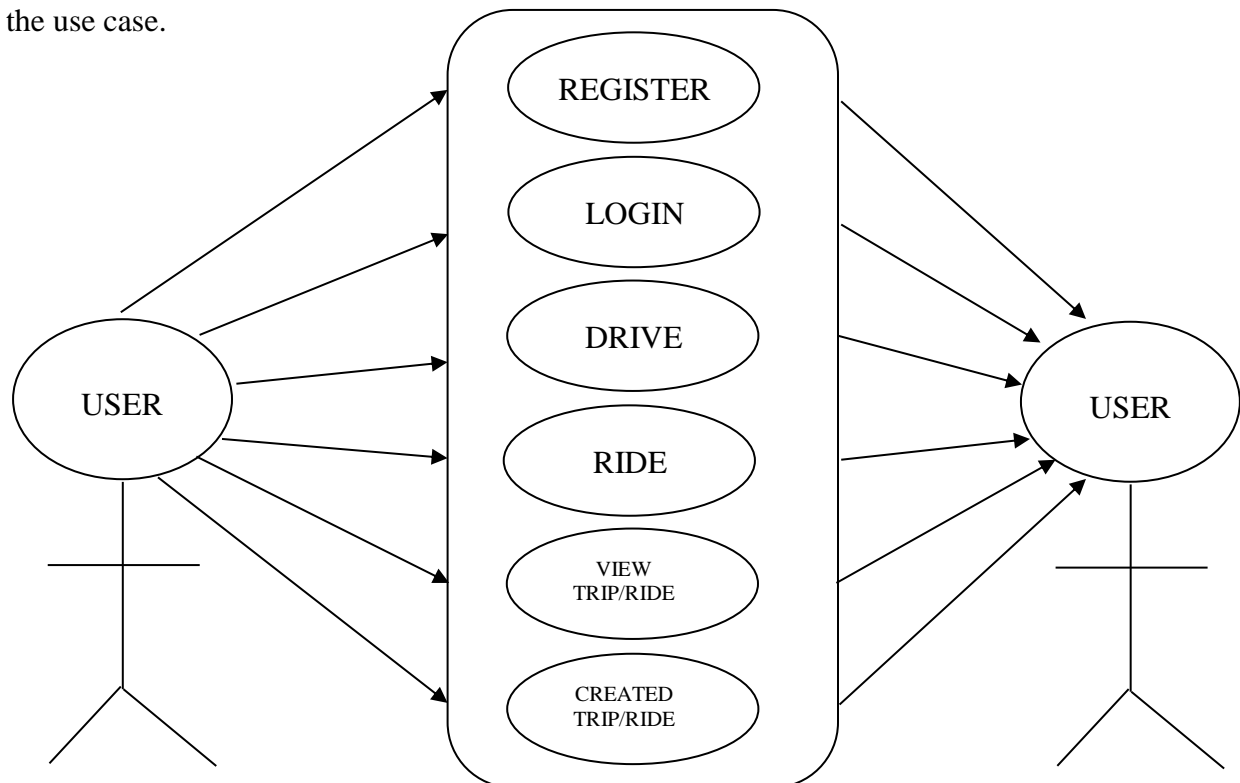
A use case is a description of a system's behavior from user's standpoint. For system developers, that is a valuable tool. A use case diagram is a graph of actors, a set of use case enclosed by a system boundary, communication and association between the actors. Use case model represent the functionality of a system or a class manifested to external interaction with the system.

USE CASE: A use case describes a sequence of action that provide something of measurable value to an actor and is drawn as a horizontal ellipse

ACTORS: An actor is a person, organization, or external system that play a role in one or more interaction with your system.

ASSOCIATIONS: Association between actors and use case are indicated in use case diagram by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Association are modeled as a line connecting use cases and actors to one another,

With an optional arrow head on one end of the line. The arrow head is often used for indicating the direction of the initial invocation of the relationship or to indicate the primary actors within the use case.



3.5 INPUT DESIGN

In the input design, user-oriented inputs are converted into a computer based system format. It also includes determining the record media, method of input, speed of capture and entry on to the screen. Online data entry accepts commands and data through a keyboard. The major approach to input design is the menu and the prompt design. In each alternative, the user's options are predefined. The data flow diagram indicates logical data flow, data stores, source and destination. Input data are collected and organized into a group of similar data. Once identified input media are selected for processing. Once required data are identified appropriated input are identified. The input medias used in this project are keyboard and mouse.

3.6 OUTPUT DESIGN

It is the part of overall system design. The goal of the output design is to capture the output and get the data into format suitable for the computer. Data flow diagram identifies the data tone captured and the output to the system. One of the important features of an information system for users is the output it produces. Output is the information delivers to the users delivered to the users through the information system. Without quality output the entire system appears to be unnecessary that users will avoid using it. Users generally merit the system solely by its output in order to create the most useful output possible. One works closely with the user through an interactive process, until the result is considered to be satisfactory.

SYSTEM ENVIRONMENT

CHAPTER 4

SYSTEM DEVELOPMENT

SYSTEM ENVIRONMENT

An important matter in building an application is selecting hardware and software. The hardware drives the software to facilitate solutions. Factors like cost performance and reliability etc are taken into consideration during the purchase of the hardware component for any computerized system.

4.1 INTRODUCTION

In computer software, an operating environment or integrated applications environment is the environment in which users run application software. The environment consists of a user interface provided by an applications manager and usually an application programming interface (API) to the applications manager.

An operating environment is usually not a full operating system but is a form of middleware that rests between the OS and the application. For example, the first version of Microsoft Windows, Windows 1.0, was not a full operating system, but a GUI laid over DOS albeit with an API of its own. Similarly, the IBM U2 system operates on both Unix/Linux and Windows NT. Usually the user interface is text-based or graphical, rather than a command-line interface (e.g., DOS or the Unix shell), which is often the interface of the underlying operating system.

4.2 HARDWARE REQUIREMENTS

Every business enterprise maintains large volume of data for its operations. Normally with the traditionally methods of storing data and information fails the chances that data loss its integrity.

- Processor : Intel Pentium dual core/above
- RAM : 2 GB or more
- Hard Disk Drive : 80 GB or more
- Monitor : 14.1-inch color monitor (LCD, CRT or LED)
- Clock speed : 2.24GHz

4.2 SOFTWARE REQUIREMENTS

- Operating system : Microsoft Windows XP/above
- Platform : WAMP/ XAMPP
- Front End : PHP
- Back End : MySQL SERVER

4.3 TOOLS AND PLATFORM

In software engineering, the terms front end and back refers to the separation of concerns between the presentation layer (front end), and the data access layer (back end) of a piece of software, or the physical infrastructure or hardware. In the client-server model, the client is usually considered the back end, even when some presentation work is actually done on the server.

4.4.1 FRONT END TOOL

PHP is a dynamically typed general purpose scripting language which can be embedded in HTML pages. It was designed in 1995 for implementing dynamic web pages, its name initially standing for Personal Home Pages. PHP now stands for the recursive acronym PHP: Hypertext Preprocessor. While predominantly used for server-side scripting PHP can be used for writing command line scripts and client-side GUI applications. It may also be embedded into host applications to provide them with scripting functionality. The main implementation of PHP is free open source software. This provides the de facto definition of the syntax and semantics for the language since there is no formal specification. While the end user interface and extensions API for the PHP interpreter are well documented [2, 10], scant documentation exists for the internals of the parser and the Zend engine (which interprets scripts). The lexical analyzer is defined using a Flex description of 2000 lines, and the parser a Bison specification of 900 lines. It is difficult to extract any formal specification from either source file because of: unnecessary redundancy and poor structuring of the grammars; convoluted rules which attempt to enforce static checks; embedded semantic actions trying to ensure associativity is appropriately maintained at certain points during parsing; and the requirement for tight coupling between the lexical analyzer and parser due to in-string syntax which is not amenable to processing using the traditional lexical analysis and parser interface.

4.4.2 BACK END TOOL

MY-SQL SERVER

MYSQL is open source relational database management system (RDBMS). Its name is a combination of “My”, the name of co-founder Michael Widenius’s daughter, and “SQL.”, the abbreviation for Structured Query Language.

MY SQL is free and open-source software under the term of GNU General public License, and is also available under a variety of proprietary licenses. My-SQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystem (now Oracle corporation). In 2010 when oracle acquired Sun, Widenius forked the open-source MYSQL project to create MariaDB

MySQL is a component of the LAMP web application software stack (and others), Which is an acronym for the LINUX, APACHE, MySQL, PERL/ PHP / PYTHON. MySQL is used by many database-driven web applications, including DRUPAL, JOOMLA and WordPress. MySQL is also used by many popular websites including Facebook, Flickr etc.

FEATURES OF MY-SQL SERVER

Deployment

MySQL can be built and installed manually from source code, but it is most commonly installed from a binary package unless special customizations are required. On most Linux distributions, the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

LAMP software bundle, displayed here together with Squid. Though MySQL began as a low-end alternative to more powerful proprietary database, it has gradually evolved to support higher scale needs as well. It is still most commonly used in small to medium scale single-server deployment, either as a component in a LAMP-based web application or a standalone database server. Much of MySQL appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tool such as PhpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory

Backup software

MySQL dump is a logical backup tool included with both community and enterprise edition of MySQL. It supports backing up from all storage engine. MySQL Enterprise Backup is a hot utility included as a part of the MySQL Enterprises subscription from Oracle, offering native InnoDB hot backup, as well as backup for other storage engines

XtraBackup is an open-source MySQL hot backup software program. Features include hot, non-locking, backup for InnoDB storage, incremental backup, streaming parallel-compressed backup, throttling based on the number of I/O operation per second, etc.

High availability software

Oracle MySQL offers a high availability solution with a mix of tool including the MySQL router and the MySQL shell. They are based on Group Replication, open source tools.

4.2.3 Operating System

The Windows operating system for desktop PCs are more formally called Microsoft Window and is actually a family of operating system for personal computer. Windows dominates the personal computer world by some estimates, more than 90 percent of all personal computer – the remainder running Linux and Mac operating systems.

Window provides a graphical user interface (GUI), virtual memory management, multitasking and support for many peripheral devices. In addition to windows operating system for personal computers, Microsoft also offers operating system for server and mobile devices .

WINDOWS 10

Windows 10 is Microsoft windows successor to Windows 8.Windows 10 debuted on July 29,2015 ,following a “ technical preview” beta release of the new operating system that arrived in Fall and a “consumer preview “beta in early 2015 .Microsoft claims Windows 10 features fast start up and resume ,built –in security and the return of the Start Menu in an expanded form .This version of Windows will also feature Microsoft Edge ,Microsoft new browser .Any qualified device(such as a tablet ,PC’s ,smartphone and Xbox consoles)can be upgraded to Windows 10 ,including those with pirated copies of windows .

SYSTEM IMPLEMENTATION

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled it can cause chaos and confusion.

5.2 IMPLEMENTATION LOGIC

Implementation includes all those activities that take place to convert from the old system to new system. The new system may be totally new, replacing an existing manual or automated system or it may be a major modification to an existing system. Proper implementation is essential to provide a reliable system to meet the organization requirements. Successful implementation may not guarantee improvement in the organization using the system, but improper installation will prevent it.

The implementation stage involves the following tasks:

- ✓ Careful planning
- ✓ Investigation of system and constraints
- ✓ Design of method to achieve the changeover phase
- ✓ Training of staffs in the changeover phase
- ✓ Evaluation of the changeover method

The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and same time users are trained in the new procedures.

Implementation Procedures:

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system, people who are not sure that the software is meant to make their job easier. In the initial stage, they doubt about the software but we have to ensure that the resistance does not build up as one has to make sure that

- ✓ The active user must be aware of the benefits of using the system.
- ✓ Their confidence in the software is built up.
- ✓ Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server the actual process won't take place.

SAMPLE CODING

LOGIN.PHP

CODE:

```
<?php
$msg = "";
require_once "model/db.php";
if (isset($_POST['SUBMIT'])) {
    $username = $_POST['username'];
    $passwrđ = $_POST['passwrđ'];
    $qry = "SELECT username, passwrđ, user_id, mobile, mobile_verified, first_name, last_name
    FROM registration WHERE username='" . $username . "' AND passwrđ='" . $passwrđ . "' LIMIT
    1";
    $result = mysqli_query($con, $qry);
    if (mysqli_num_rows($result) == 1) {
```

```

while ($r = mysqli_fetch_array($result)) {
$_SESSION['user_id'] = $r['user_id'];
$_SESSION['mobile'] = $r['mobile'];
$_SESSION['mobile_verified'] = $r['mobile_verified'];
$_SESSION['name'] = $r['first_name'].' '.$r['last_name'];
}
if (isset($_SESSION['user_id'])) {
if (!headers_sent()) {
header('location:drive.php');
} else {
echo '<script>location.href="drive.php"</script>';
}
}
}
else {
echo "<script>alert('Invalid Username or Password')</script>";
}
mysqli_close($con);
}
?>

```

REGISTRATION.PHP

CODE:

```

<?php
require_once "model/db.php";
if(isset($_POST['SUBMIT']))
{
$first_name = $_POST['first_name'];

```

```

$last_name = $_POST['last_name'];
$email = $_POST['email'];
$mobile = $_POST['mobile'];
$username = $_POST['username'];
$password = $_POST['password'];
$qry = "INSERT INTO registration (first_name, last_name, email, mobile, username, passwd) VA
LUES ('".$first_name."', '".$last_name."', '".$email."', '".$mobile."', '".$username."', '".$passw
ord."')";
mysqli_query($con, $qry);
$uid = mysqli_insert_id($con);
if($uid>0){
$_SESSION['user_id'] = $uid;
$_SESSION['mobile'] = $mobile;
require_once 'send-otp.php';
$res = send_otp($mobile);
echo '<style> .smodal { display: block; } </style>';
}
}
mysqli_close($con);
?>

```

DRIVE.PHP

CODE:

```

<?php
require_once "model/db.php";
if (!isset($_SESSION['user_id'])) {
if (!headers_sent()) {
header('location:login.php');

```



```
} else {  
echo '<script>location.href="login.php"</script>';  
}  
}  
?>
```

```
<?php  
if($_SESSION['mobile_verified']==0){  
$mobile = $_SESSION['mobile'];  
require_once 'send-otp.php';  
$res = send_otp($mobile);  
echo '<style> .smodal { display: block; } </style>';  
include_once 'modal-otp.php';  
}  
?>
```

```
<?php  
require_once "model/db.php";  
if(isset($_POST['SUBMIT']))  
{  
$user_id = $_SESSION['user_id'];  
$source = $_POST['source'];  
$destination = $_POST['destination'];  
$year = $_POST['year'];  
$month = $_POST['month'];  
$day = $_POST['day'];
```

```

$date = $year."-".$month."-".$day;
$hours = $_POST['hours'];
$minutes = $_POST['minutes'];
$seconds = 00;
$time = $hours.":".$minutes.":".$seconds;
$vehicle = $_POST['vehicle'];
if(strtotime($date.' '.$time)>time()){
$qry = "INSERT INTO trip (source, destination, datee, timee, vehicle_id, user_id) VALUES ('".$source."', '".$destination."', '".$date."', '".$time."', $vehicle, $user_id)";
mysqli_query($con, $qry);
$ins = mysqli_insert_id($con);
if($ins>0)
{
echo "<script>alert('Details Saved Successfully')</script>";
}
else
{
echo "<script>alert('Something went wrong. Please try again later')</script>";
}
}
else
{
echo "<script>alert('Please select a valid date')</script>";
}
}
mysqli_close($con);
?>

```

RIDE.PHP**CODE:**

```
<?php
require_once "model/db.php";
if (!isset($_SESSION['user_id'])) {
if (!headers_sent()) {
header('location:login.php');
} else {
echo '<script>location.href="login.php"</script>';
}
}
?>
```

```
<?php
if ($_SESSION['mobile_verified'] == 0) {
$mobile = $_SESSION['mobile'];
require_once 'send-otp.php';
$res = send_otp($mobile);
echo '<style> .smodal { display: block; } </style>';
include_once 'modal-otp.php';
}
?>
```

```
<?php
require_once "model/db.php";
if (isset($_POST['SUBMIT'])) {
$user_id = $_SESSION['user_id'];
```

```

$source = $_POST['source'];
$destination = $_POST['destination'];
$year = $_POST['year'];
$month = $_POST['month'];
$day = $_POST['day'];
$date = $year . "-" . $month . "-" . $day;
$hours = $_POST['hours'];
$minutes = $_POST['minutes'];
$seconds = 00;
$time = $hours . ":" . $minutes . ":" . $seconds;
if (strtotime($date . ' ' . $time) > time()) {
    $qry = "INSERT INTO ride (source, destination, datee, timee, user_id)
VALUES ('" . $source . "', '" . $destination . "', '" . $date . "', '" . $time . "', '" . $user_id . "')";
    mysqli_query($con, $qry);
    $ins = mysqli_insert_id($con);
    if ($ins > 0) {
        echo "<script>alert('Details Saved Successfully')</script>";
    } else {
        echo "<script>alert('Something went wrong. Please try again later')</script>";
    }
    } else {
        echo "<script>alert('Please select a valid date')</script>";
    }
    }
    mysqli_close($con);
?>

```

REQUEST.PHP**CODE:**

```

<?php
require_once "model/db.php";
if($_REQUEST['action']=='verify_otp')
{
if($_SESSION['otp']== $_REQUEST['otp'])
{
$qry="Update registration set mobile_verified = 1 where user_id =". $_SESSION['user_id'];
mysqli_query($con, $qry);
$_SESSION['mobile_verified'] = 1;
echo json_encode(array('success'=>1, 'page' => 'drive.php'));
}
else
{
echo json_encode(array('msg'=>'Incorrect OTP','success'=>0));
}
die;
}
if($_REQUEST['action']=='resend_otp'){
require_once 'send-otp.php';
$mobile = $_SESSION['mobile'];
$res = send_otp($mobile);
if($res)
echo json_encode(array('msg'=>'OTP Sent Successfully'));
die;
} ?>

```

DB.PHP**CODE:**

```
<?php
$con = mysqli_connect("localhost", "root", "", "cargo");
session_start();
?>
```

SEND-OTP.PHP**CODE:**

```
<?php
function send_otp($mobile) {
    $mobileNumber = '91' . $mobile;
    $otp = mt_rand(99999, 999999);
    $_SESSION['otp'] = $otp;
    $otp_msg = "Dear Customer, Please enter the One Time Password (OTP) to continue CarGo Car
    pool Registration. Your OTP verification code is " . $otp;
    $authKey = "306471AeHKiEVz5de3b4df";
    $senderId = "xCARGO";
    $message = urlencode($otp_msg);
    $url = "http://api.msg91.com/api/sendotp.php?authkey=" . $authKey . "&mobile=" . $mobileNu
    mber . "&message=" . $otp_msg . "&sender=" . $senderId . "&otp=" . $otp;
    $ch = curl_init();
    curl_setopt_array($ch, array(
        CURLOPT_URL => $url,
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_POST => true,
    ));
```

```

curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
$output = curl_exec($ch);
if (curl_errno($ch)) {
    echo 'error:' . curl_error($ch);
}
curl_close($ch);
return $output;
}
?>

```

CREATED DRIVES.PHP

CODE:

```

<?php
require_once "model/db.php";
if (!isset($_SESSION['user_id'])) {
    if (!headers_sent()) {
        header('location:login.php');
    } else {
        echo '<script>location.href="login.php"</script>';
    }
}
?>

```

```

<?php
$query = "SELECT * FROM trip INNER JOIN vehicle ON trip.vehicle_id = vehicle.vehicle_id WHERE t
rip.user_id = " . $_SESSION['user_id'] . " order by trip.trip_id desc";
$result = mysqli_query($con, $query);

```

```
$count = mysqli_num_rows($result);
?>
```

```
<?php
if ($count > 0) {
$i = 0;
?>
```

```
<?php
while ($res = mysqli_fetch_array($result)) {
$i++;
?>
```

```
<?php
echo date('M d, Y h:i A', strtotime($res['datee'] . ' ' . $res['timee']));
?>
```

CREATED RIDES.PHP

CODE:

```
<?php
require_once "model/db.php";
if (!isset($_SESSION['user_id'])) {
if (!headers_sent()) {
header('location:login.php');
} else {
```



```

echo '<script>location.href="login.php"</script>';
}
}
?>

```

```

<?php
$qry = "SELECT * FROM ride WHERE user_id =" . $_SESSION['user_id'] . " order by ride_id desc";
$result = mysqli_query($con, $qry);
$count = mysqli_num_rows($result);
?>

```

```

<?php
if ($count > 0) {
    $i = 0;
?>

```

```

<?php
while ($res = mysqli_fetch_array($result)) {
    $i++;
?>

```

```

<?php
echo date('M d, Y h:i A', strtotime($res['datee'] . ' ' . $res['timee']));
?>

```

VIEW REQUESTS.PHP

CODE:

```

require_once "model/db.php";

```

```

if (!isset($_SESSION['user_id']) || !isset($_REQUEST['id'])) {
if (!headers_sent()) {
header('location:login.php');
} else {
echo '<script>location.href="login.php"</script>';
}
}

```

```

<?php
include_once 'menu.php';
$sqlqry = "SELECT COUNT(p_id) as p_count FROM passenger WHERE status=1 AND trip_id=" . $_REQUEST['id'];
$result2 = mysqli_query($con, $sqlqry);
$res2 = mysqli_fetch_array($result2);
?>

```

```

<?php
$qry = "SELECT passenger.p_id, registration.first_name, registration.last_name, registration.email, registration.mobile, trip.source, trip.destination, trip.datee, trip.timee, passenger.status, vehicle.vehicle_name, vehicle.passengers_no FROM passenger INNER JOIN trip ON trip.trip_id = passenger.trip_id INNER JOIN ride ON ride.ride_id = passenger.ride_id INNER JOIN registration ON ride.user_id = registration.user_id INNER JOIN vehicle ON vehicle.vehicle_id = trip.vehicle_id WHERE passenger.trip_id=" . $_REQUEST['id'] . " AND trip.user_id=" . $_SESSION['user_id'] . " order by passenger.p_id ASC";
$result = mysqli_query($con, $qry);
$count = mysqli_num_rows($result);
?>

```

```
<?php
```

```
if ($count > 0) {
```

```
$i = 0;
```

```
?>
```

```
<?php
```

```
while ($res = mysqli_fetch_array($result)) {
```

```
$i++;
```

```
?>
```

```
<?php
```

```
if (isset($_REQUEST['btn_request'])) {
```

```
$p_id = $_REQUEST['hdn_p_id'];
```

```
$qry = "UPDATE passenger SET status=1 WHERE p_id=" . $p_id;
```

```
mysqli_query($con, $qry);
```

```
$ins = mysqli_affected_rows($con);
```

```
if ($ins > 0) {
```

```
echo "<script>alert('Request Approved Successfully')</script>";
```

```
echo "<meta http-equiv=refresh content=\"0; URL=" . $_SERVER['REQUEST_URI'] . "\">";
```

```
} else {
```

```
echo "<script>alert('Something went wrong. Please try again later')</script>";
```

```
}
```

```
}
```

```
if (isset($_REQUEST['btn_reject'])) {
```

```
$p_id = $_REQUEST['hdn_p_id'];
```

```
$qry = "UPDATE passenger SET status=2 WHERE p_id=" . $p_id;
```

```
mysqli_query($con, $qry);
```

```
$ins = mysqli_affected_rows($con);
```

```

if ($ins > 0) {
echo "<script>alert('Request Rejected')</script>";
echo "<meta http-equiv=refresh content=\"0; URL=\" . $_SERVER['REQUEST_URI'] . \">";
} else {
echo "<script>alert('Something went wrong. Please try again later')</script>";
}
}
?>

```

VIEW_TRIPS.PHP

CODE:

```

<?php
require_once "model/db.php";
if (!isset($_SESSION['user_id']) || !isset($_REQUEST['id'])) {
if (!headers_sent()) {
header('location:login.php');
} else {
echo '<script>location.href="login.php"</script>';
}
}
?>

```

```

<?php
$qry = "SELECT * FROM ride WHERE ride_id = " . $_REQUEST['id'];
$result = mysqli_query($con, $qry);
$res0 = mysqli_fetch_array($result);
$qry = "SELECT * FROM trip INNER JOIN vehicle ON trip.vehicle_id = vehicle.vehicle_id INNER JO
IN registration ON trip.user_id = registration.user_id WHERE trip.datee = '" . $res0['datee'] . '"

```

```
AND trip.source = " . $res0['source'] . " AND trip.destination = " . $res0['destination'] . " ORDE
R BY trip.timee asc";
```

```
$result = mysqli_query($con, $qry);
$count = mysqli_num_rows($result);
?>
```

```
<?php
if ($count > 0) {
    $i = 0;
    ?>
```

```
<?php
while ($res = mysqli_fetch_array($result)) {
    $i++;
    $qry1 = "SELECT * FROM passenger INNER JOIN ride ON passenger.ride_id = ride.ride_id INNER
JOIN registration ON registration.user_id = ride.user_id WHERE passenger.trip_id = " . $res['trip
_id'] . " AND registration.user_id = " . $_SESSION['user_id'];
    $result1 = mysqli_query($con, $qry1);
    $count1 = mysqli_num_rows($result1);
    $res1 = mysqli_fetch_array($result1);
    ?>
```

```
<?php
echo date('M d, Y h:i A', strtotime($res['datee'] . ' ' . $res['timee']));
?>
```

```
<?php
if($count1 == 0)
```

```
{
?>
<button type="submit" id="btn_request" name="btn_request">Send Request</button>
<?php
}
else{
if ($res1['status'] == 0) {
?>
<div class="alert alert-info" role="alert">
Approval Pending
</div>
<?php
} else if ($res1['status'] == 1) {
?>
<div class="alert alert-success" role="alert">
Request Accepted
</div>
<?php
} else if ($res1['status'] == 2) {
?>
<div class="alert alert-danger" role="alert">
Request Rejected
</div>
<?php
}}
?>

<?php
```

```

if (isset($_REQUEST['btn_request'])) {
    $trip_id = $_REQUEST['hdn_trip_id'];
    $ride_id = $_REQUEST['hdn_ride_id'];
    $qry = "INSERT INTO passenger (trip_id, ride_id, status) VALUES ($trip_id, $ride_id, 0)";
    mysqli_query($con, $qry);
    $ins = mysqli_insert_id($con);
    if ($ins > 0) {
        echo "<script>alert('Request Sent Successfully')</script>";
        echo "<meta http-equiv=refresh content=\"0; URL=\"".$_SERVER['REQUEST_URI']."\">";
    } else {
        echo "<script>alert('Something went wrong. Please try again later')</script>";
    }
}
?>

```

LOGOUT.PHP

CODE:

```

<?php
require_once "model/db.php";
session_destroy();
header('location:login.php');
?>

```

SYSTEM TESTING

CHAPTER 6

SYSTEM TESTING

6.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, testing presence an interesting anomaly for the software engineer.

Testing Objectives includes:

- ✓ Testing is a process of executing a program with the internet of finding an error.
- ✓ A good test case is one that has a probability of finding an as yet undiscovered error.
- ✓ A successful test is one that uncovers an undiscovered error.

Testing Principles:

- ✓ All tests should be traceable to end user requirements.
- ✓ Tests should be planned long before testing begins.
- ✓ Testing should begin on a small scale and progress towards testing in large.
- ✓ Exhaustive testing is not possible.

Testing phase of system development life cycle tests system design. Testing of system decides whether the newly designed system works properly or not. After the development of documentation manually about the system this stage is checked. And if the system working properly then it will be considered for implementation and if isn't the system analyst is informed to find out generated errors or problems and to find out its solution. This process is known as debugging.

6.2 UNIT TESTING

Unit testing is a software development process in which the smallest testable part of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated. In computer programming, unit testing is a software testing method by which individual unit of source code, sets of one or more computer program module together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

The goal of unit is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy.

6.3 INTEGRATION TESTING

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These “design items”, i.e. Assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulates usage of shared data areas and inter-process communication is tested and individual subsystem are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing. The overall idea is a “building block” approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.

Some different types of integration testing are big bang, top-down, and bottom-up. Other Integration Patterns [1] are: Collaboration Integration, Backbone Integration, Layer Integration, Client/Server Integration, Distributed Services Integration and High-frequency Integration.

6.3 SYSTEM TESTING

System testing is a level of the software testing where a complete and integrated software is tested. The purpose of this test is to evaluated the system’s compliance with the specified requirement. System testing of the software or hardware is testing conducted in complete, integrated system to evaluate the system’s compliance with its specified requirement. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

6.3.1 TEST PLANNING AND CASES

A test plane is a document detailing the objectives, target market, internal beta team, and processes for a specific beta test for a software or hardware product. The plan typically contains a detailed understanding of the eventual workflow.

SYSTEM MAINTENANCE

CHAPTER 7

SYSTEM MAINTENANCE

The maintenance phase of the software cycle is the time in which a software product performs useful works. After a system is successfully implemented, it should be maintained in proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make it adaptable to the changes in the system environment. There may be social, technical and other environment changes, which affect a system, which is implemented. Software product enhancement may involve providing new functional capabilities, improving user displace and mode of interaction, upgrading the performance characteristics of the system. So only through proper system maintenance procedures, the system can be adapted to cope up with these changes.

Software maintenance is of course, far more than “Finding mistakes”. We may define maintenance by describing four activities that are undertaken to after a program is released of use.

The **first maintenance activity** occurs because it is unreasonable to assume that software testing will uncover latent errors in a large software system. During the use of any large programs, errors will occur and report to the developer. The process that includes the diagnosis and correction of one more errors is called corrective maintenance.

The **second activity** that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspects of computing. Therefore, adapting maintenance-an activity that modifies software to properly interface with a changing environment is both necessary and common place.

The **third activity** that may apply to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions and general enhancements are received from user. To satisfy requests in this category, perfective maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

The **fourth maintenance activity** occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

FUTURE ENHANCEMENT AND SCOPE OF FURTHER DEVELOPMENT

CHAPTER 8

DEVELOPMENT FUTURE ENHANCEMENT AND SCOPE OF FURTHER

Accuracy:

It is one of the features of computer that it provides accuracy result. It is often said, “Man does mistakes but machines never”. When a man works there seems number of errors and mistakes in performing calculation and other things, while it is not in case with computer.

Security:

The security is maintained very comfortably in computers. Secrets are not disclosed off, as there are password settings in the computers. So the firm can get the benefit of non-disclosure of its secrets to the outsider competitors as well as other non-authorized persons.

Time saving:

The computerized system is time saving. On an immediate inquiry, the operator has to just click some switches and get the information in few seconds.

Perform Repetitive Task Very Well:

It is said that computer never tires. It does the repetitive task very well as it never feels laziness. Performance of repetitive task by a man is not well performed, as man tires very quickly and feels lazy. It is not in the case with computer.

Low cost:

The computer need not change year as it happens in case of registers. In registers system clerk, employs and also managers are needed to fulfil the management activity while in computer system the two persons are enough for all the work. So in proposed system the two persons are enough for all the work.

Easy Maintenance:

The maintenance of the computer is not tough job as compared to the registers, as it may not be lost or damaged. So, there is only soft or easy maintenance needed with computers.

Easy modification:

In case of any change in data stored, the modifications can be done very easily and clearly through computers, but this not possible with registers.

Attractive and Status:

Computer creates different kinds of impression in mind of computer or any other person. Computer tends to increase in reputation/status of the firm. Whenever party required any information the units has just to print to out clicking some switches. So with the help of computers, promoters can increase their profit.

Hence, with these merits, it would be very unwise and reckless to manage the organization through the system of bookkeeping. As it is said that prevention is better the cure, it would be very fine and intelligent decision to have computer application system in management.

LIMITATION OF THE SYSTEM

1. No option for changing password if forgotten
2. Since the Google map API's are not used in our project the system only compares riders and driver who share same source and destination that is a person cannot the trip in its way point.
3. No chat interface between rider and driver

FUTURE SCOPE OF APPLICATION

The system can be improved in future by adding a chat interface between driver and rider. Here we created a web application called "CARGO" and in future we can expand it as a mobile application which can be used on both IOS and ANDROID platforms.

CONCLUSION

CHAPTER 9

CONCLUSION

After the completion of the project the problems in the existing system would overcome. The CARGO process made computerized to produce human errors and to increase the efficiency. The main focus of the project is to lessen human efforts. The maintenance of the records is made efficient, as all the records are stored in the MS ACCESS database, through which data can be retrieved easily. The navigation control is provided in all the forms to navigate through the large amount of records. If the no. of record is very large then user has to just type the customer Id and user gets the result immediately. The vehicle and customer are given a particular unique id customer Id so that they can be accessed correctly and without errors. Our main aim of the project is to get the correct information about a particular member who s hiring the vehicle.

The problem, which existed in the earlier system, has been removed to a large extend and it is expected that this project will go a long way in satisfying users' requirements. The rent a car management system will improve not only the efficiency but also reduce human stress threr by indirectly improving car hiring.

BIBLIOGRAPHY

CHAPTER 10

BIBLIOGRAPHY

Book References:

1. **High Performance MySQL** - Baron Schwartz, Derek J. Balling, Jeremy Zawodny, and Peter Zaitsev
2. **Analysis and Design of Information Systems-Second Edition** - McGraw-Hill International Edition James A Senn
3. **Software Engineering – A Practitioner’s Approach** - Roger S Pressman
4. **System Analysis and Design** - Galgotia Publications (P) Ltd,Elias M Awad

Web References:

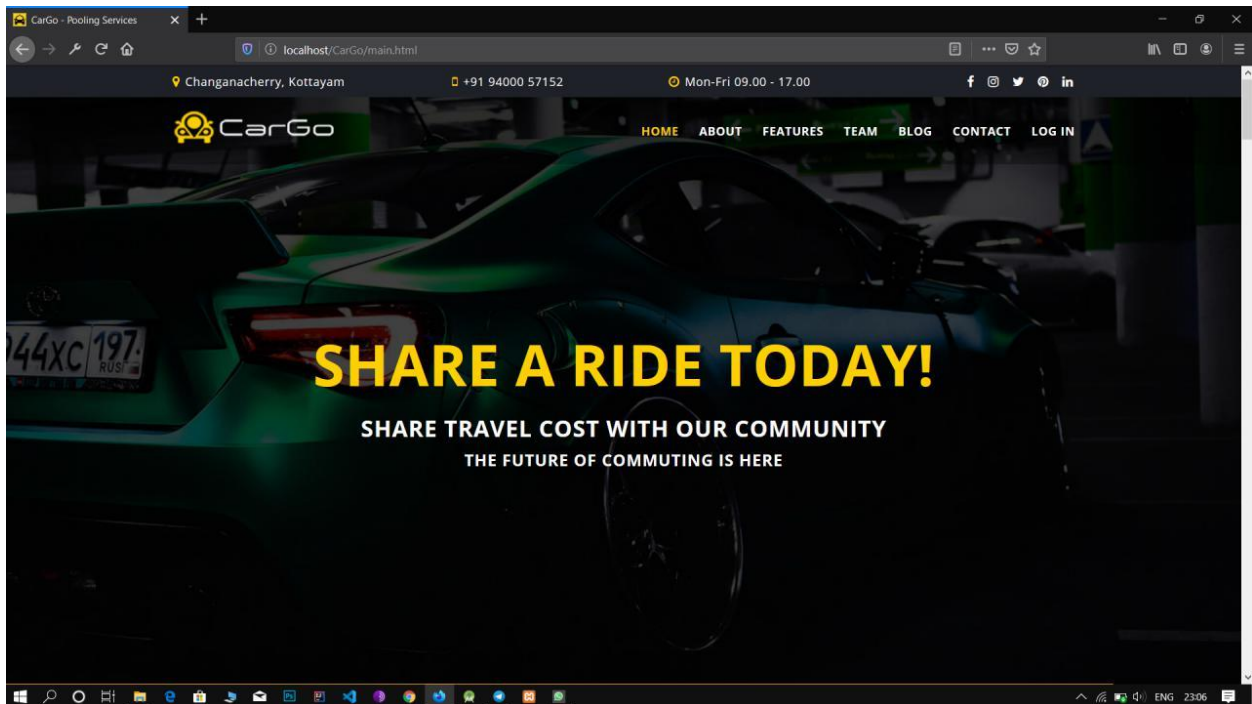
1. <http://www.stackoverflow.com>
2. <http://www.w3schools.com>
3. [http:// developer.mozilla.org](http://developer.mozilla.org)
4. <http://tutorialspoint.com>

APPENDIX

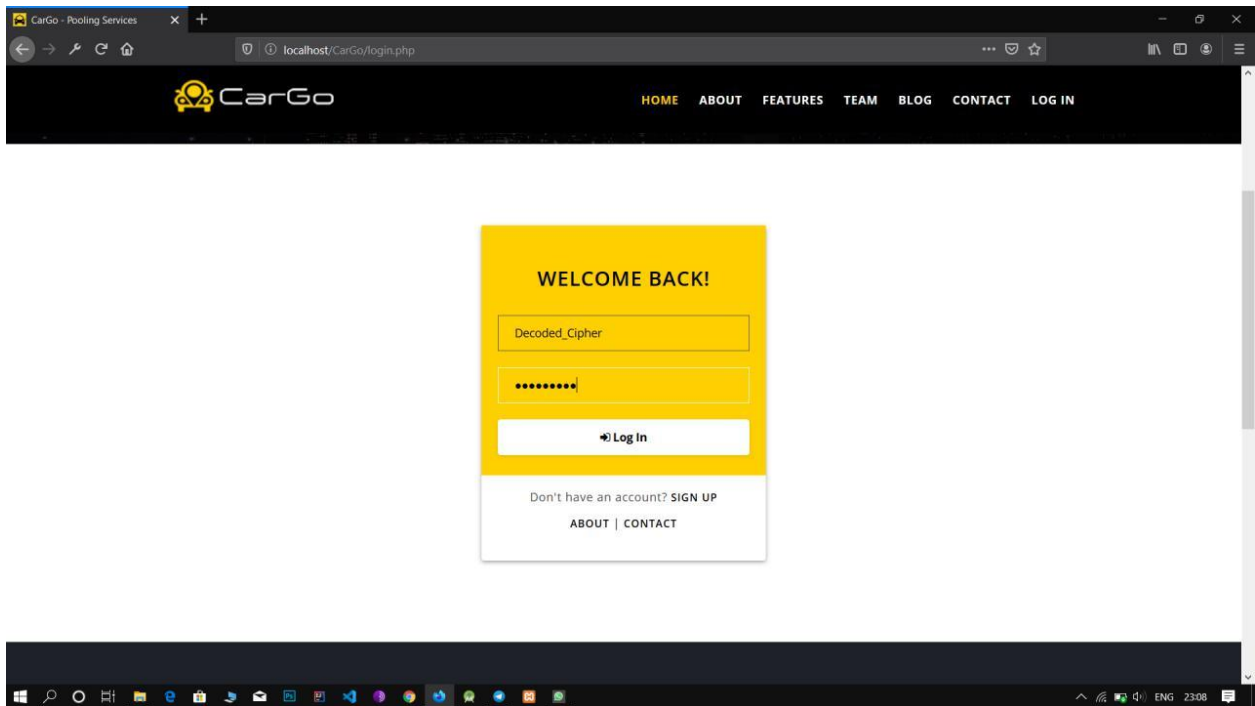
CHAPER 11

APPENDIX

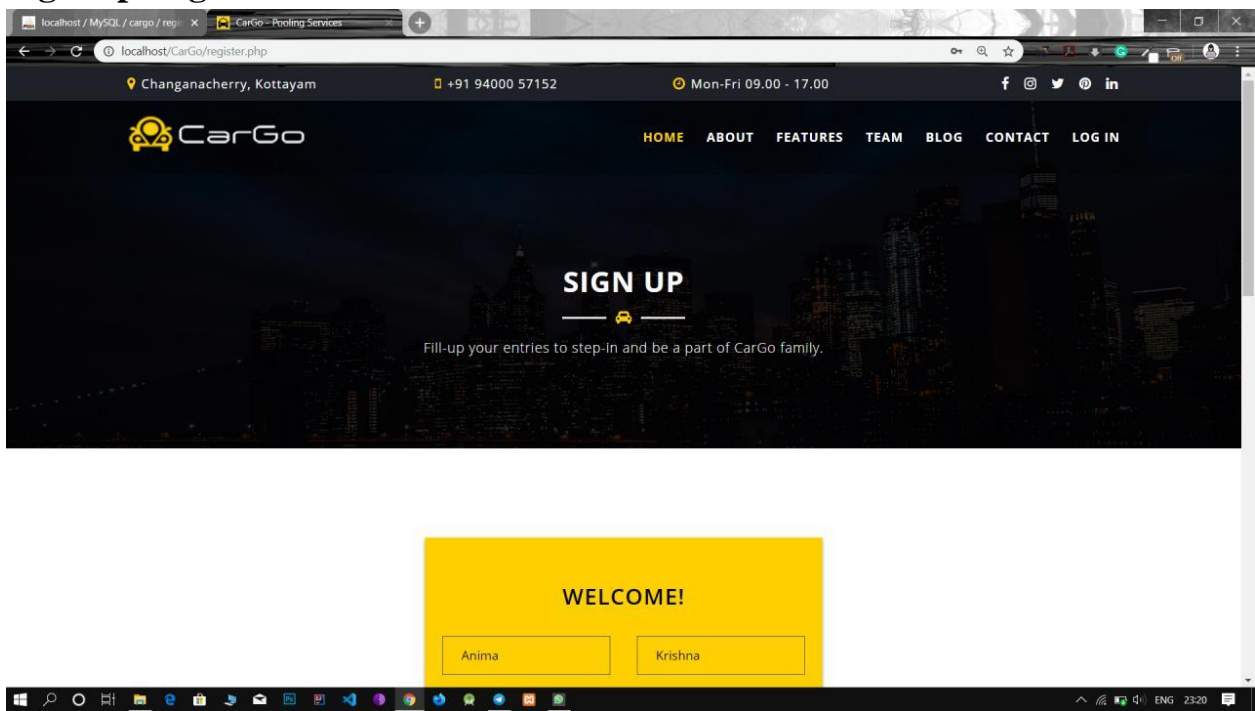
Home Page: MAIN.HTML

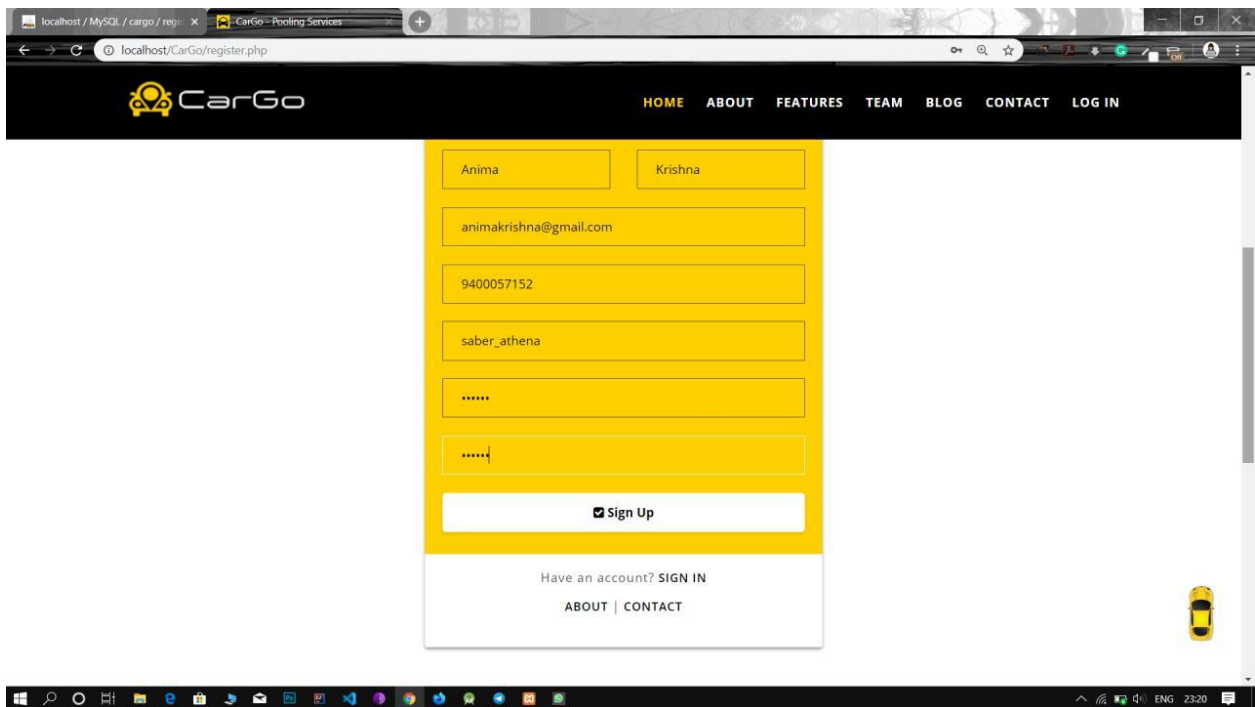


Sign-In Page: LOGIN.PHP



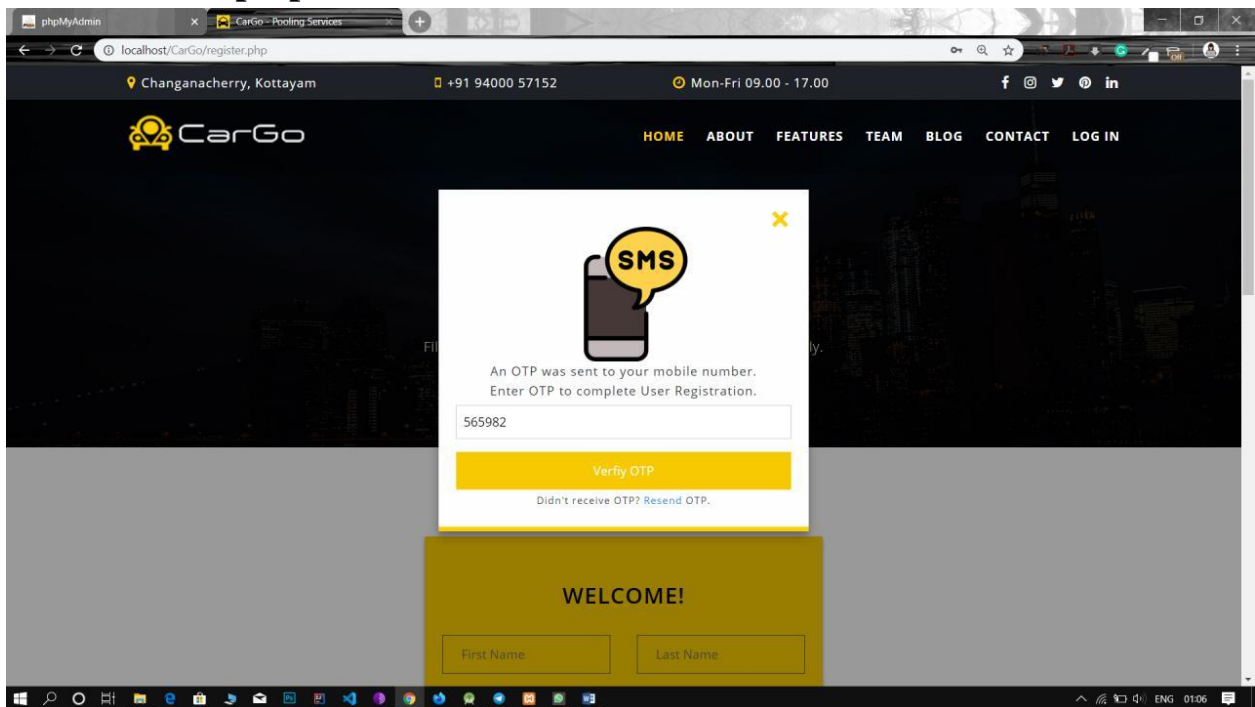
Sign-Up Page: REGISTER.PHP





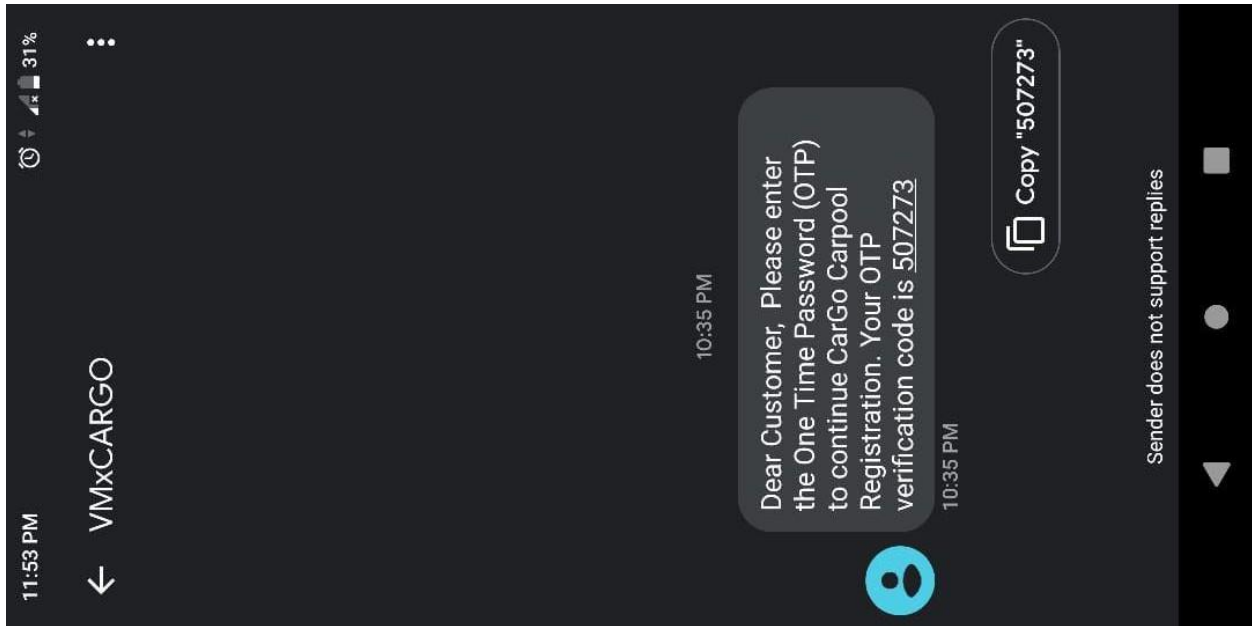
The screenshot shows a web browser window with the URL `localhost/CarGo/register.php`. The page has a dark header with the CarGo logo and navigation links: HOME, ABOUT, FEATURES, TEAM, BLOG, CONTACT, and LOG IN. The registration form is a yellow box with the following fields: First Name (Anima), Last Name (Krishna), Email (animakrishna@gmail.com), Phone Number (9400057152), Username (saber_athena), Password (masked with dots), and Confirm Password (masked with dots). Below the fields is a 'Sign Up' button. At the bottom of the form, there is a link to 'SIGN IN' for existing users and links for 'ABOUT' and 'CONTACT'. A small yellow car icon is visible in the bottom right corner of the page.

Enter OTP Pop-up:

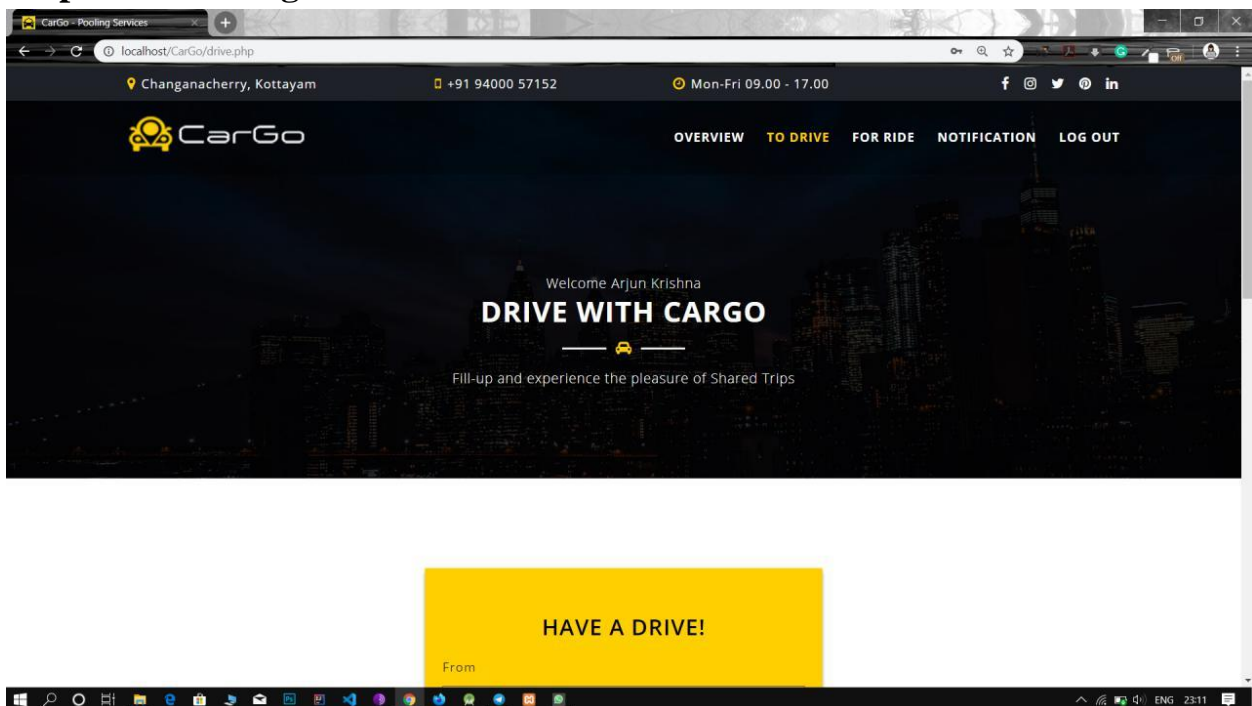


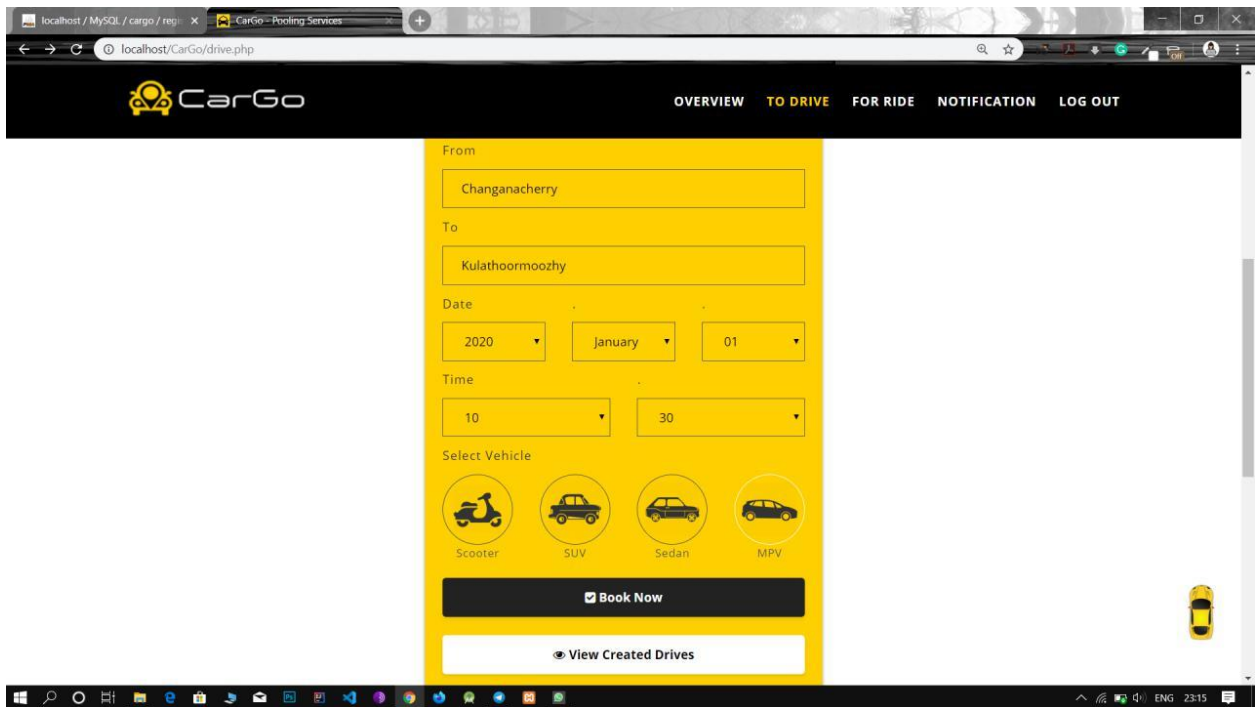
The screenshot shows the same CarGo registration page, but with an OTP verification pop-up displayed in the center. The pop-up has a white background and a yellow border. It features an 'SMS' icon with a speech bubble, the text 'An OTP was sent to your mobile number. Enter OTP to complete User Registration.', a text input field containing the number '565982', a yellow 'Verify OTP' button, and a link 'Didn't receive OTP? Resend OTP.'. Below the pop-up, a yellow box with the text 'WELCOME!' and input fields for 'First Name' and 'Last Name' is visible. The background of the registration page is dark and shows a map of the area around Changanacherry, Kottayam. The browser window shows the URL `localhost/CarGo/register.php` and the page title 'CarGo - Pooling Services'.

OTP Verification for Mobile Phones:



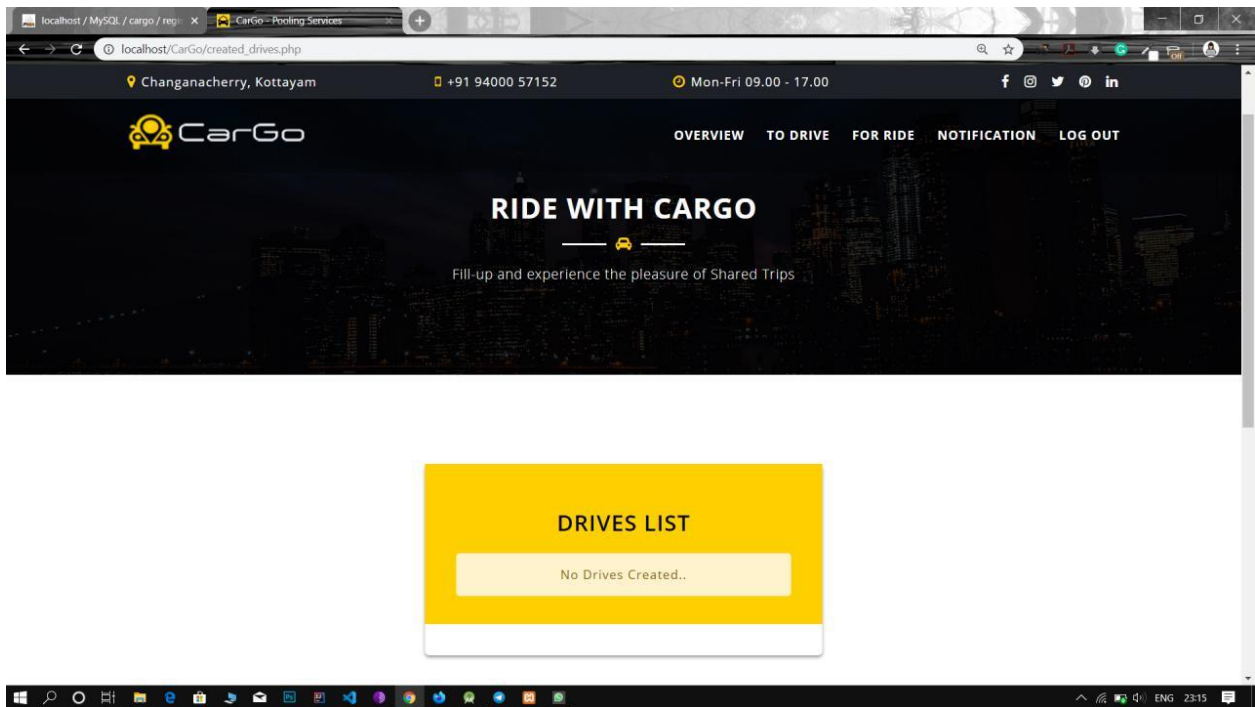
Trip Creation Page: DRIVE.PHP

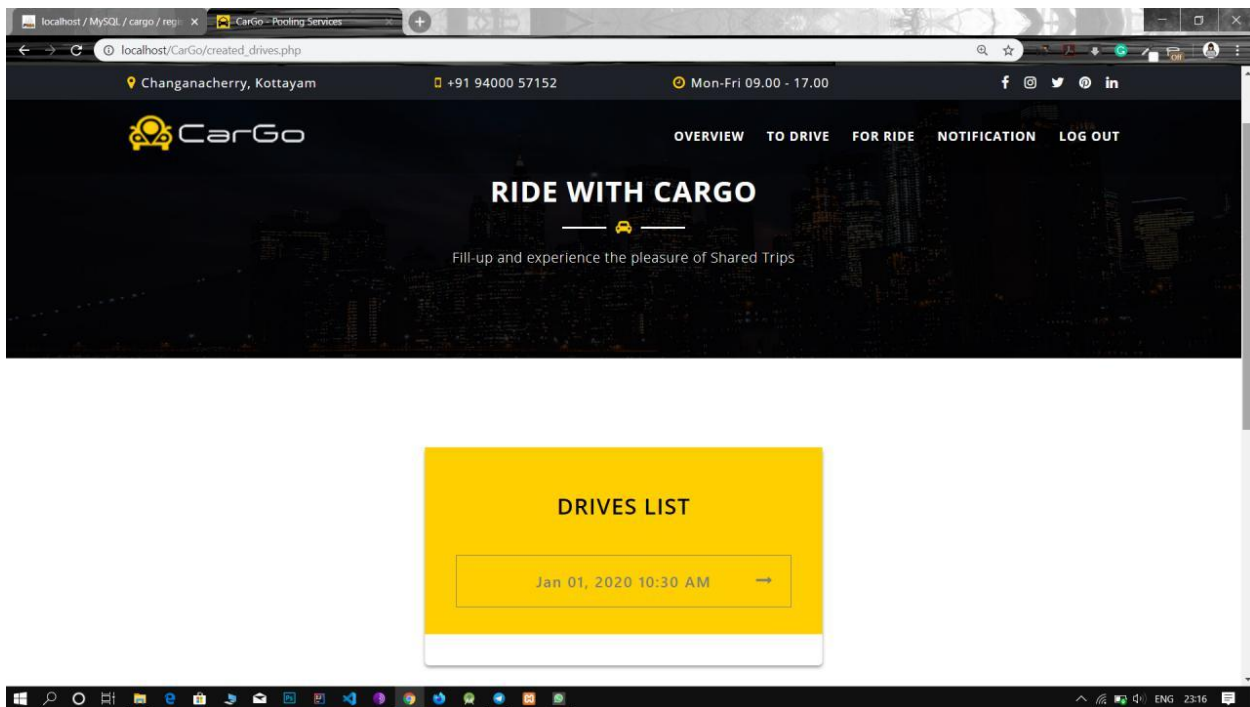




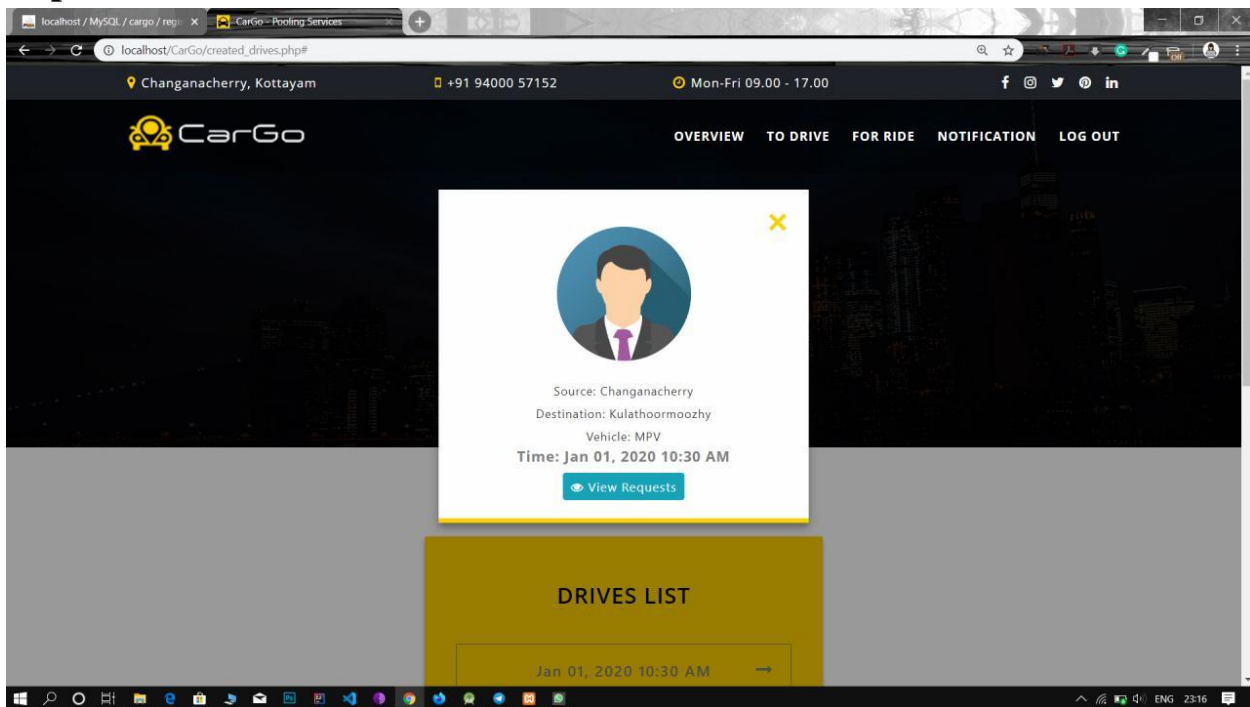
The screenshot shows a web browser window with the URL `localhost/CarGo/drive.php`. The page has a black header with the CarGo logo and navigation links: OVERVIEW, TO DRIVE, FOR RIDE, NOTIFICATION, and LOG OUT. The main content area is a yellow form for booking a ride. It includes fields for 'From' (Changanacherry), 'To' (Kulathoormoozhy), 'Date' (2020, January, 01), and 'Time' (10, 30). Below these are four vehicle options: Scooter, SUV, Sedan, and MPV, each with a corresponding icon. A 'Book Now' button is at the bottom of the form, and a 'View Created Drives' link is below it. A small yellow car icon is visible on the right side of the page.

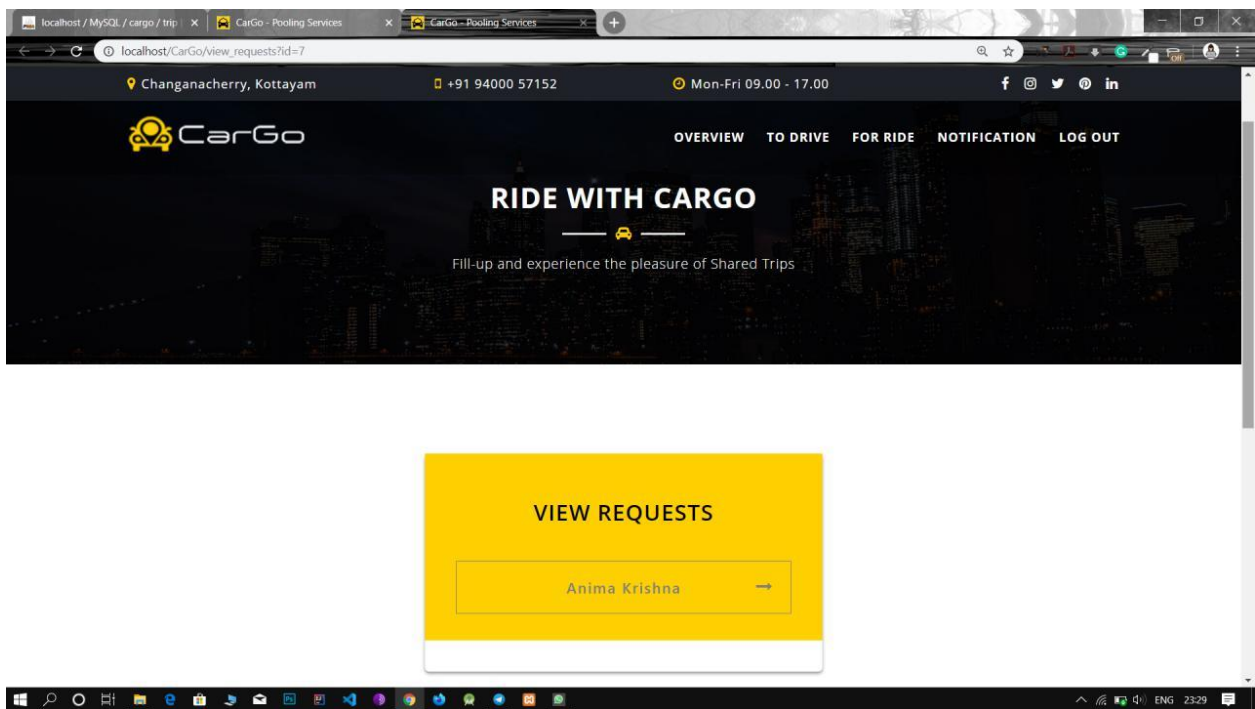
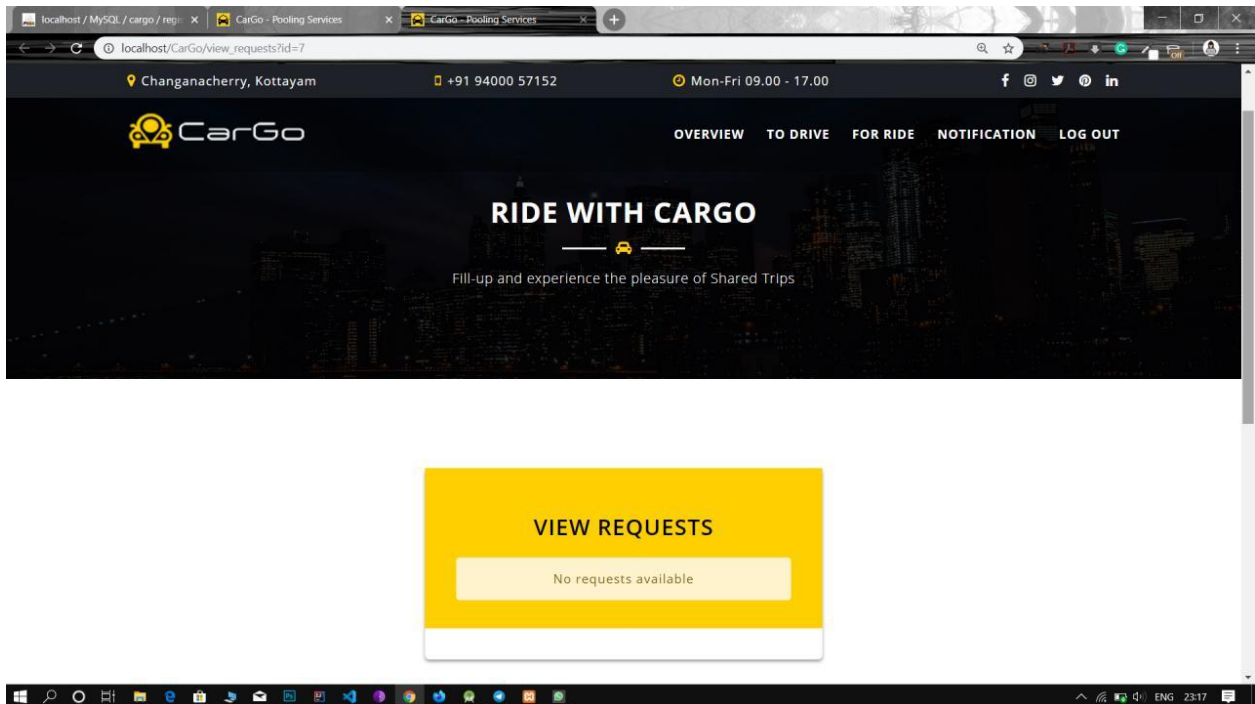
Result: CREATED_DRIVES.PHP



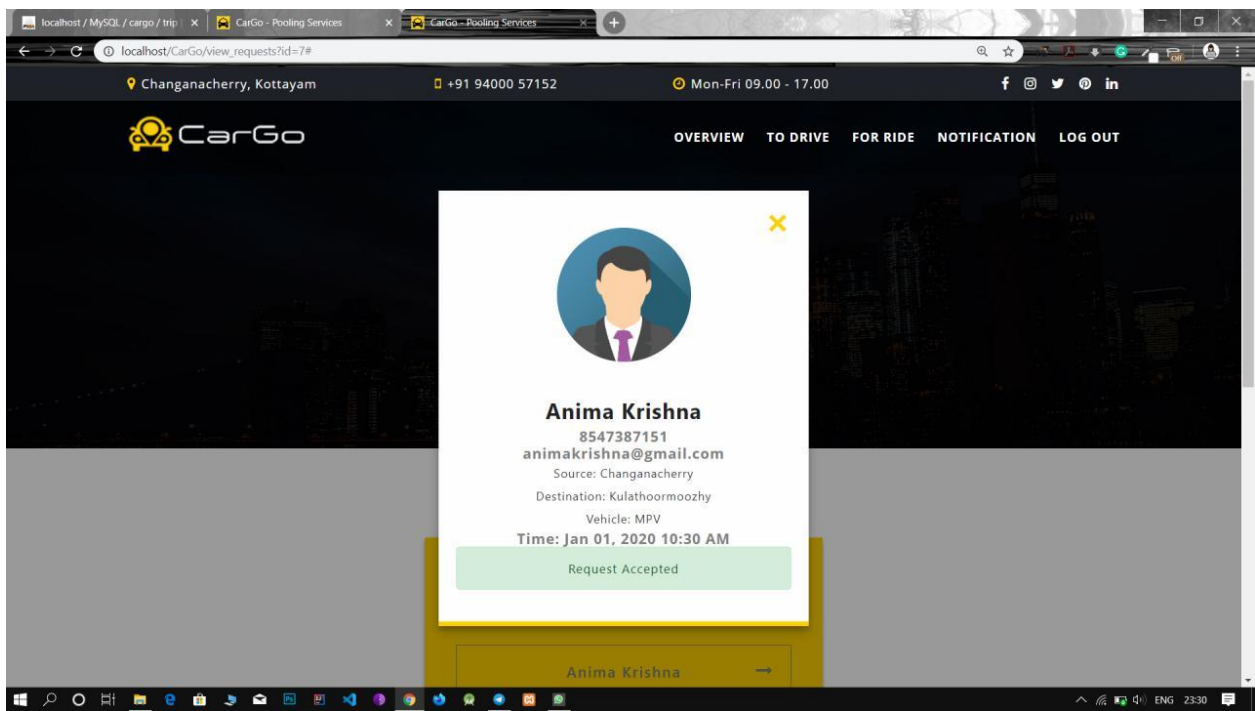
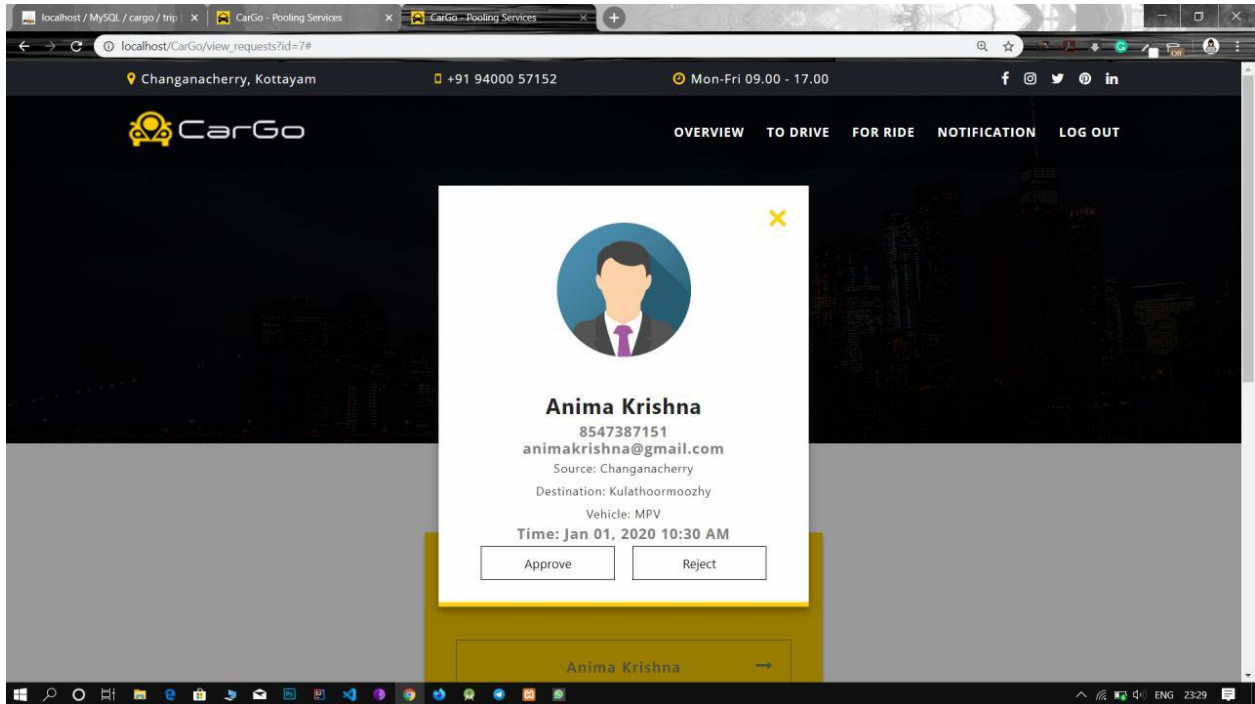


Expanded Result: CREATED_DRIVES.PHP

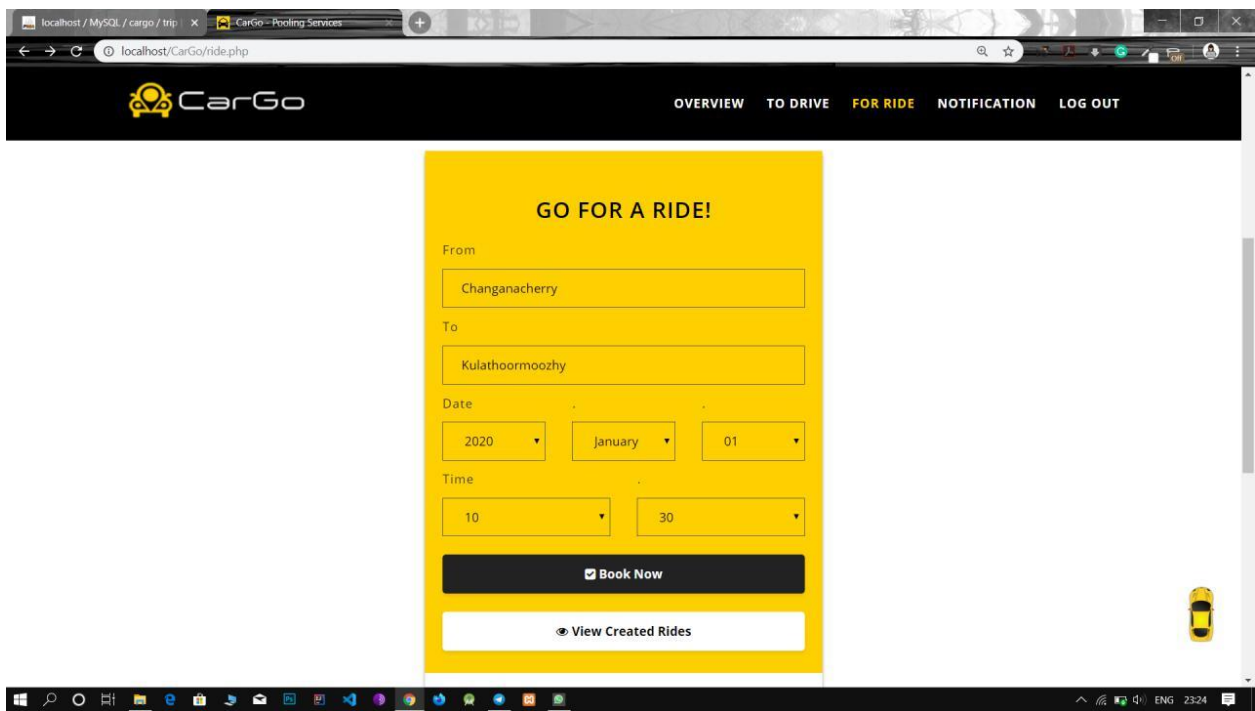
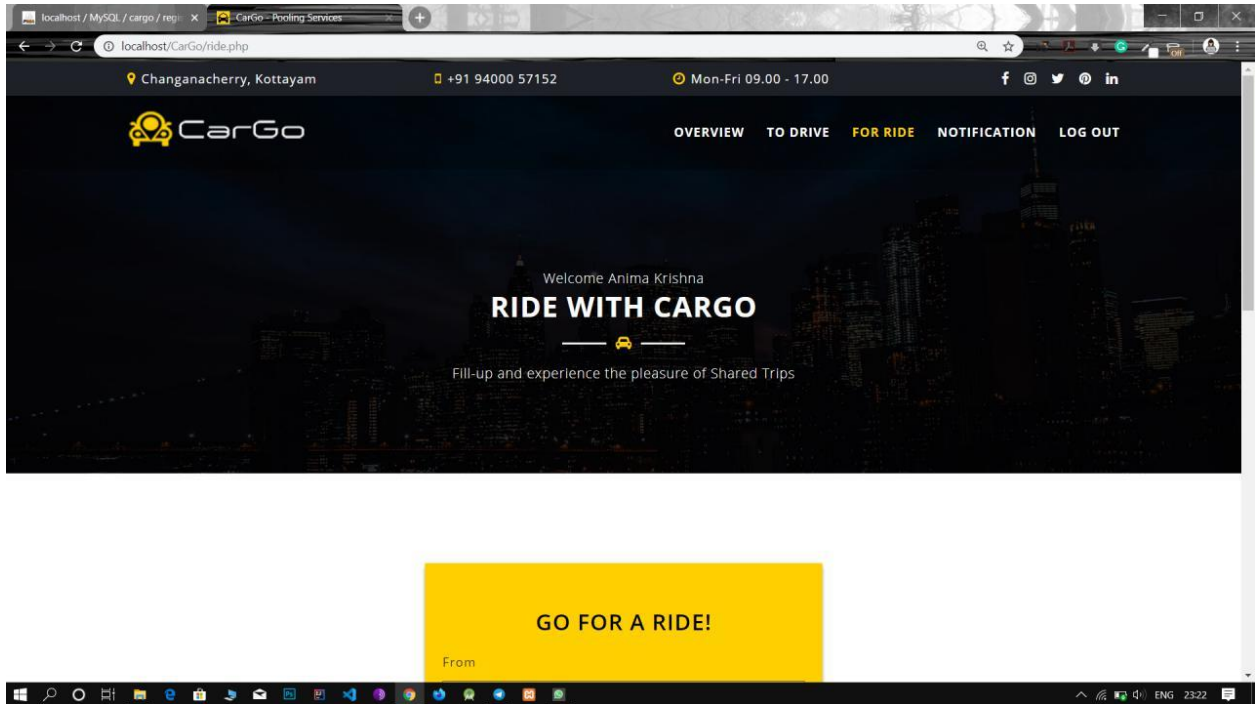


Result: VIEW_REQUESTS.PHP

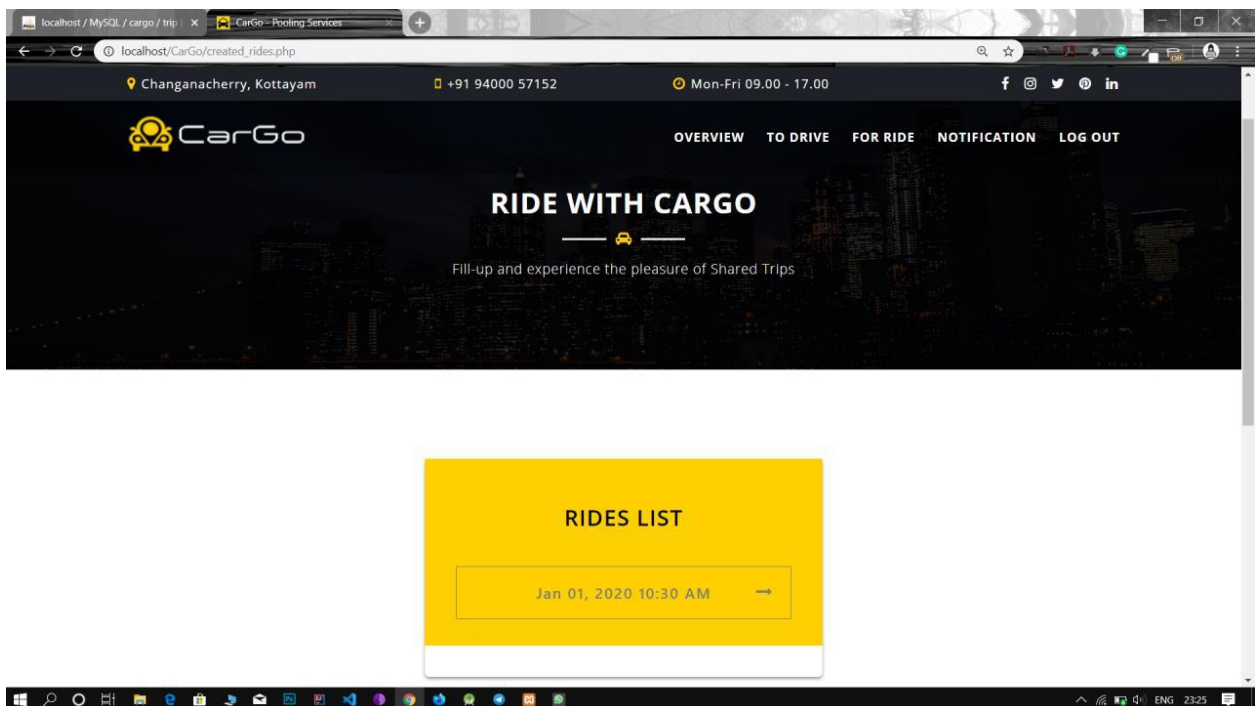
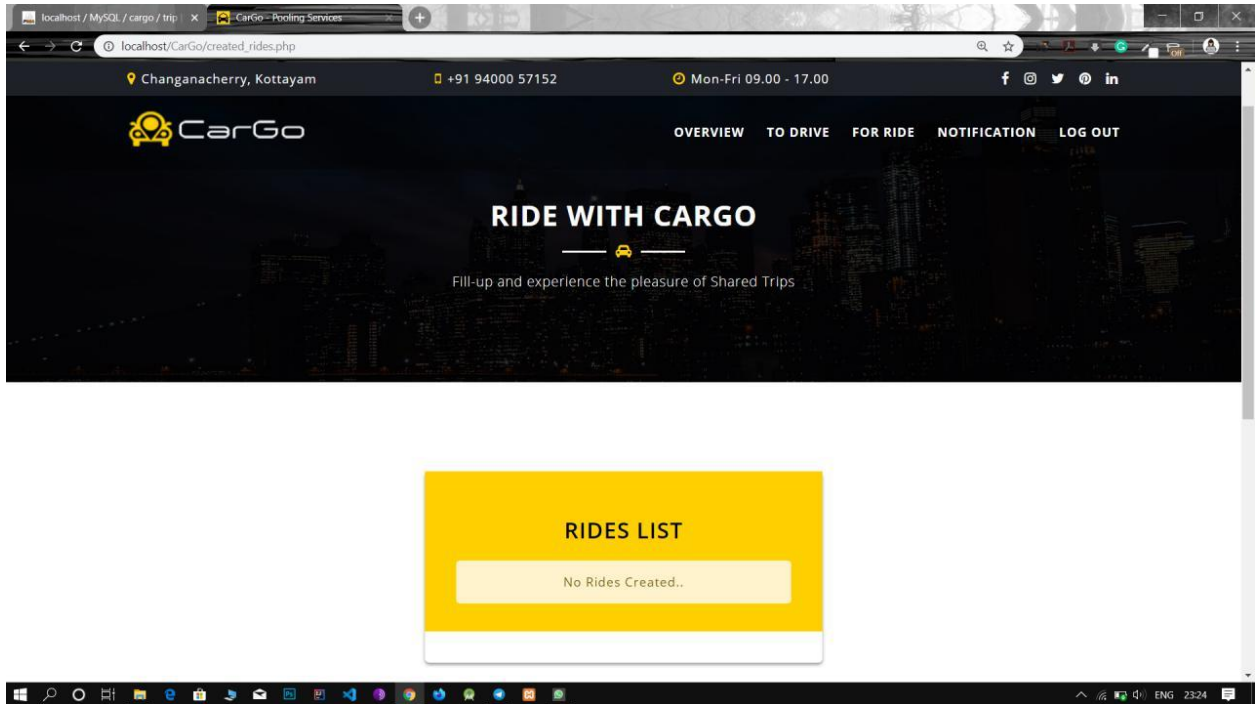
Expanded Result: VIEW_REQUESTS.PHP



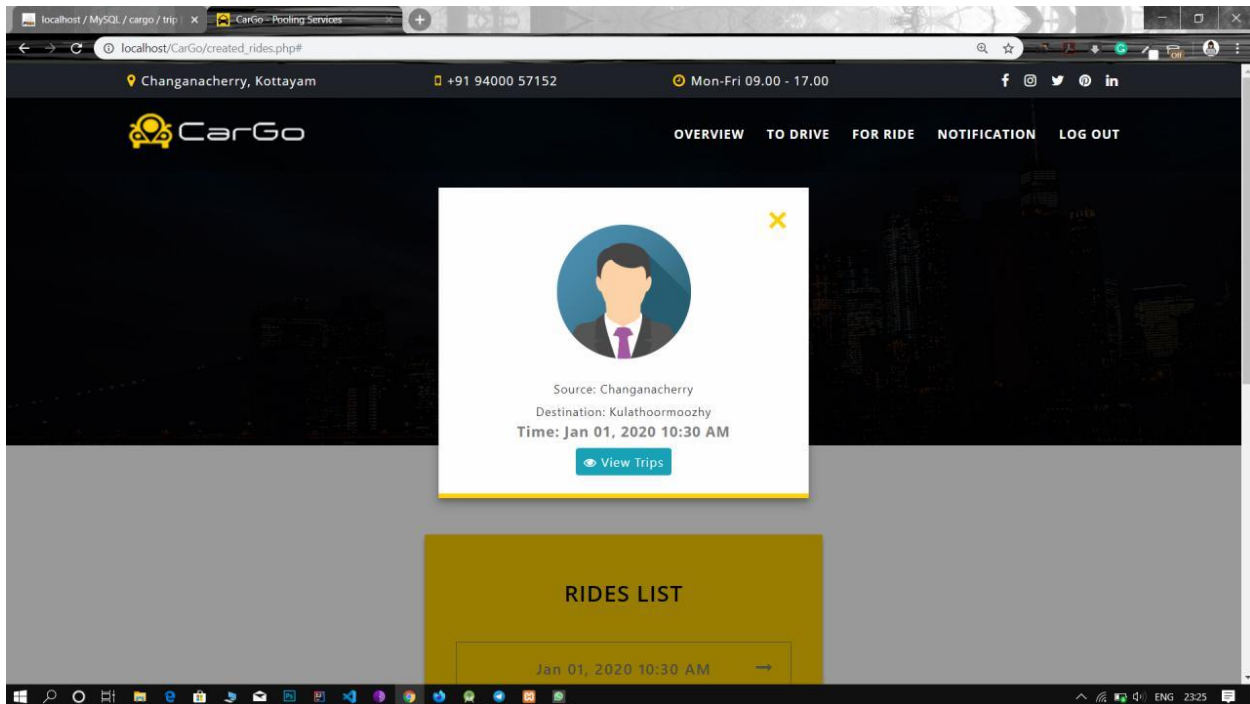
Ride Joining Page: RIDE.PHP



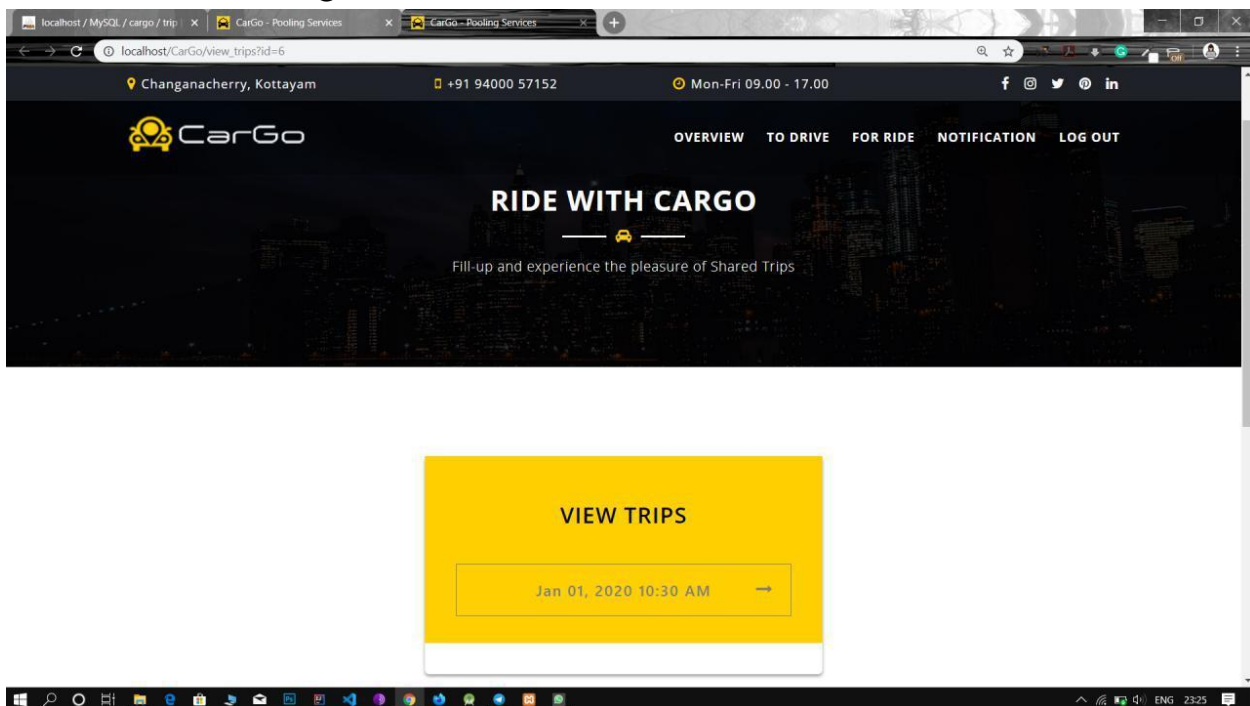
Result: CREATED_RIDES.PHP



Expanded Result: CREATED_RIDES.PHP



Result: VIEW_REQUESTS.PHP



Expanded Result: VIEW_TRIPS.PHP

