

# **PARABELLO**

Project Report Submitted by

**MUHAMMED SALEEL**

**(KIARBCA007)**

Under the guidance of

**Mrs. SURAYYA A**

(Assistant Professor, Department of Computer Science)

In partial fulfilment of the requirements

For the award of the degree of

**BACHELOR OF COMPUTER APPLICATION**

Of the Calicut University



**DEPARTMENT OF COMPUTER SCIENCE**

**MES KALLADI COLLEGE**

**MANNARKKAD, PALAKKAD**

**678583**

## **CERTIFICATE**

This is to certify the project worked entitled “PARABELLO” is a bonafide record of work done by **MUHAMMED SALEEL (KIARBCA007)** submitted on partial fulfilment of the requirements for award of the degree of **BACHELOR OF COMPUTER APPLICATION** of the Calicut University under my supervision.

**Signature of Guide**

**Date:**

**Counter Signed by**

**HOD**

**Principal**

External viva-voice conducted on:.....

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **DECLARATION**

I **Mrs. SURAYYA.A** hereby declare that the project entitled “**PARABELLO**” submitted to the Calicut University in the partial fulfilment of the requirement for the award of degree of **BACHELOR OF COMPUTER APPLICATION** is a record of original work done by **MUHAMMED SALEEL (KIARBCA007)** during his period of study at **MES KALLDI COLLEGE**, under the guidance of me.

**Place: MANNARKKAD**

**Date:**

Signature of guide

**SURAYYA.A**

Assistant Professor

Department of computer science

**MES KALLADI COLLEGE**

**MANNARKKAD**

## **DECLARATION**

I **MUHAMMED SALEEL (KIARBCA007)** hereby declare that the project entitled “**PARABELLO**” submitted to the Calicut University in partial fulfilment of the requirement for the award of degree of **BACHELER OF COMPUTER APPLICATION**, is a record of original work done by during my period of study at **MES KALLADI COLLEGE** Under the guidance of Mrs. Surayya A, Assistant Professor, Department of Computer Science.

**Place: MANNARKKAD**

**Signature of the Student**

**Date:**

**MUHAMMED SALEEL**

## ACKNOWLEDGEMENT

The success of the project depends upon the effort invested. It's my responsibility to acknowledge and thank the individuals who has contributed in the successful completion of this project. I wish to express my heartfelt gratitude to **Mr.Jaleel T.K, Principal, MES Kalladi College** for the encouragement and inspiring guidance throughout preparation of this project.

I express my deep sense of gratitude and sincere thanks to the **HOD Mrs. Preetha Rajagopalan M, Department of Computer Science**, for her kind suggestion and co-operation. I am very much thankful to **Mrs. Surayya A, Assistant Professor, Department of Computer Science**, for his valuable guidance and support as a guide and tutor to successfully complete this project.

I am also thankful to our department faculties for their continuous motivation and support for the successful completion of my project.

I wish to express my love and respect to my parents for their support, contribution and encouragement which helped me a lot to complete this project successfully.

I am very much thankful to my friends for their support and help to complete this project successfully.

**MUHAMMED SALEEL**

| <b>No</b>  | <b>INDEX</b>                               | <b>P.No</b> |
|------------|--|-------------|
| <b>1.</b>  | <b>1INTRODUCTION</b>                       | <b>7</b>    |
| <b>2.</b>  | <b>SYSTEM ANALYSIS</b>                     | <b>8</b>    |
| <b>3.</b>  | <b>FEASIBILITY STUDY</b>                   | <b>10</b>   |
| <b>4.</b>  | <b>SOFTWARE REQUIRMENT SPECIFICATION</b>   | <b>11</b>   |
| <b>5.</b>  | <b>SYSTEM DESIGN AND DEVELOPMENT</b>       | <b>18</b>   |
| <b>6.</b>  | <b>SYSTEM IMPLEMENTATION &amp; TESTING</b> | <b>24</b>   |
| <b>7.</b>  | <b>SYSTEM SECURITY MEASURES</b>            | <b>28</b>   |
| <b>8.</b>  | <b>CONCLUSION</b>                          | <b>31</b>   |
| <b>9.</b>  | <b>SCOPE FOR FUTURE WORKS</b>              | <b>32</b>   |
| <b>10.</b> | <b>APPENDIX</b>                            | <b>33</b>   |
| <b>11.</b> | <b>BIBLIOGRHY</b>                          | <b>43</b>   |
| <b>12.</b> | <b>SAMPLE SCREEN FORMATS</b>               | <b>43</b>   |

# **1. INTRODUCTION**

Intelligent profiling is the application of selling, buying and comparing cars also includes rental services. It involves selling vehicles, renting vehicle for a specific time, identifying its target buyers, comparing with desired cars, and coordinating the technical aspects before selling the car.

The main part of intelligent profiling system is finding the suitable buyers for seller. Selling and buying are made directly which reduce the risk of fraudulent activities. Buyers can contact with the genuine users and make a purchase directly and also compare the features with desired vehicles. It's helpful for buyers to find the right vehicle. Public users can use the system to comparing vehicle. The main objective of the Intelligent Profiling System is to manage the details of Buyers, Seller, Selling vehicle and Renting vehicle. It manages all the information about Buying, Selling, and Compare. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to make a better marketplace for buyers and sellers who desire to buy or sell a used car.

## **1.1 Project Overview**

Parabello, Intelligent Profiling System is a software project that serves the functionality of an online marketplace. The system allows only registered users to login and new users are allowed to register on the application. This is proposed to be a web application. The project provides most of the basic functionality required for buying and selling process. It allows the user to select from a list of used cars. Once the user select the desired car, the system then allows the user to select images and display the details about the car like kilometres driven fuel type, year of manufacture, etc. All this data is logged in the database and the user is given an id for making a purchase or sale.

The purpose of Parabello system is to provide a single interface for both search and compare process before buying. The process of buying and selling is usually referred to as making a deal with satisfying price. Buying include comparison option which is helpful in cross matching the desired features. Sellers details are provided only when the user will view seller profile. This application also provide option for renting vehicle by accepting required id from occupier.

## **2. SYSTEM ANALYSIS**

### **2.1 The Existing System**

This existing system is not providing secure registration and profile management of all the users properly. This system is not providing rental services. This system doesn't provide tracking of user's activities and their progress. This manual system gives us very less security for buying and some fraudulent activity due to mismanagement. This system is not providing proper seller information. The system is giving inconsistent information through the online market place.

#### **Limitation of existing system**

1. It is time consuming
2. Fraud buyers and sellers
3. It consumes lot of manpower to better results
4. It lacks of data security
5. Retrieval of data takes lot of time
6. Lack of rental service options

### **2.2 Proposed System**

The reach of computers in the industry, as in other field has been remarkable. Right from the stage of enquiry the computerized system provide sample scope of systematic and cost effective management. The scope for human error is reduced considerably and records are neat perfect and orderly. For the individuals in quest of knowledge, this is indeed a boon. The users precious time can be saved and required knowledge can be gained from any place at any moment. System is developed in PHP using My SQL as back end. This system gives a platform independent environment.

#### **Advantages**

- Easy to handle.
- Best vehicle for buyers are provided by the system.
- Clear verification.
- Less work.
- Very fast.
- Customer satisfaction.



- Effective maintenance of data.
- Less manual interaction
- Compare option.
- Cost effective.

Design phase acts as a bridge between the software requirement specification and the implementation phase, which satisfies the requirements. The aim of the project was to develop an efficient, customer friendly, high performance system with minimum storage and no redundancy. Extreme care has been given during design and development of the system.

### **3. FEASIBILITY STUDY**

The feasibility analysis deals with all the analysis that takes up in developing the project. An on the spot study relieved that there is much to be done by the way of updating the systems, providing more scope for a computerized environment in the education environment. With many systems available, we conducted a feasibility analysis to determine on the best one that will be tailor-made to suit the needs of the unit.

A system is a feasible system only if it is feasible within limited resource and time. The different feasibilities that have to analyze are:

- Technical feasibility
- Economic feasibility
- Social feasibility

#### **3.1 Technical Feasibility**

Technical feasibility canters on the existing computer and to what extend it can support the proposed addition. For e.g. the current computer is operating at 80% capacity, then running another environment. It needs a web browser, web server and the support for Microsoft SQL server connectivity, which are satisfied by internet infrastructure. So this system is technically feasible without requiring any additional hardware or software. If the budget is constraint then the project is judged technically feasible

#### **3.2 Economical Feasibility**

Economic feasibility or cost/ benefit feasibility is an assessment of the economic justification for a computer based system project. Here the proposed system is economically feasible as there are initial expenses of acquiring space in a web server is required for this system. Hence the system is economically feasible.

#### **3.3 Operational Feasibility**

This feasibility test asks if the system will work when it is developed and installed. Operational feasibility in this project: The proposed system offers greater level of user-friendliness. The proposed system produces best results and gives high performance. It can be implemented easily .So this project is operational feasible

## **4. SOFTWARE REQUIREMENT SPECIFICATION**

Software requirement specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements. Non-functional requirements impose constraints on the design or implementation such as performance engineering requirements, quality standards, or design constraints.) The specification may include a set of use cases that describe interactions the users will have with the software. The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software. On the other hand the user of the system are accessing the system through Internet and they are computer literate, so resistance from that side is also very less.

### **4.1 Functional Requirements**

Functional requirements represent the indented behaviour of the system. This behaviour may be expressed as services, tasks, or functions that the specified system is required to perform. The following functional requirements have been identified for this project

#### **4.1.1 Administrator**

This has the role of site administrator. Administrator can login into the system using username and password. He/she can handle and manage each and every activity done in the site. When they enter username and password administrator checking it with the already registered username and password in the database will validate it. The main activities include:

- Login
- Verifying buyers and sellers
- Verify selling and renting vehicle
- Add comparing car details
- Accept or Reject users
- Administrator has been seen and controls the entire site.
- Deal with the feedbacks, Complaints etc

#### **4.1.2 Seller**

This type of module can login and use the system for selling their vehicles. The main operations are

- Register and Login
- Add selling vehicle details
- Add renting vehicle details
- Insert comparison details
- View selling vehicle and update status
- View renting vehicle and update status

#### **4.1.3 Buyer**

This type of module can login and use the system for buying the desired vehicle. The main operations are

- Register and Login
- View seller list
- View renter list
- Send feedback
- Compare cars

#### **4.1.4 Public user**

This type of module can use the system for comparing vehicles. The main operations are

- Compare cars

### **4.2 Non Functional Requirements**

Non-functional requirements define the general qualities of the software product. Non-functional requirement is in effect a constraint placed on the system or on the development process.

The proposed system should be work on any windows operating system such as XP or higher platforms with no modification for adaptation. Even the fault of the platform goes for terminating the application; the current state of the system would save the partial stage wise output. It must be a robust system with useful user interaction

- **Privacy**

This system must support for three types of users. Each users must login for their functions.

- **Reliability**

This application must be reliable. This application should support the different types of service. Each service should be reliable with the user's conditions.

- **Portability**

This application must support for portable. The system must support for more than one operating system. That is the system may be installed on Linux or windows or Mac operating servers.

- **Security**

The system must support for security. This application support must offer security for user's login. That is must use a security algorithm.

- **Maintainability**

The system must support for maintainability. Every application will be maintained in future .So each system must support for maintainability.

## **4.3 Other Requirements**

### **4.3.1 PHP**

PHP is a powerful tool for making dynamic and interactive Web pages. It is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.PHP files can contains text, html tags and scripts. The PHP files returned to the browser as plain HTML. The extension of PHP files are “.PHP”, “.php3”, “.html”. PHP stands for PHP: Hypertext Pre-processor.

#### **4.3.1.1 FEATURES**

PHP is, and efficient alternative to competitors such as Microsoft's ASP.PHP files can contains text, html tags and scripts.

- PHP stands for PHP: Hypertext Pre-processor
- PHP a powerful tool for making dynamic and interactive Web pages. It is the widely used, free is a server-side scripting language, like ASP
- PHP scripts are executed on the server
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid etc)

- PHP is an open source software
- PHP is free to download and use

PHP Syntax:

PHP code is executed on the server, and the plain HTML result is sent to the browser.

### **Basic PHP Syntax:**

A PHP scripting block always starts with `<?php` and ends with `?>`. A PHP scripting block can be placed anywhere in the document. On servers with shorthand support enabled you can start a scripting block with `<?` And end with `?>`. For maximum compatibility, we recommend that you use the standard form (`<?php`) rather than the shorthand form. A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code. Each code line in PHP must end with a semicolon. The semicolon is a separator and is used to distinguish one set of instructions from one another.

There are two basic statements to output text with PHP: `echo` and `print`.

- **PHP Forms and User Input:**

The PHP `$_GET` and `$_POST` variables are used to retrieve information from forms, like user input.

- **The `$_GET` Function:**

The built-in `$_GET` function is used to collect values from a form sent with `method="get"`. Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.

- **The `$_POST` Function:**

The built-in `$_POST` function is used to collect values from a form sent with `method="post"`. Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send.

- **The PHP `$_REQUEST` Function:**

The PHP built-in `$_REQUEST` function contains the contents of both `$_GET`, `$_POST`, and `$_COOKIE`. The `$_REQUEST` function can be used to collect form data sent with both the GET and POST methods.

### **4.3.2MySQL SERVER**

MySQL is an open source, SQL Relational Database Management System (RDBMS) that is free for many uses. Early in its history, MySQL occasionally faced opposition due to its lack of support for some core SQL constructs such as sub selects and foreign keys. Ultimately, however, MySQL found a broad, enthusiastic user base for its liberal licensing terms, perky performance, and ease of use. Its acceptance was aided in part by the wide variety of other technologies such as PHP, Java, Perl, Python, and the like that have encouraged its use through stable, well-documented modules and extensions. MySQL has not failed to reward the loyalty of these users with the addition of both sub selects and foreign keys as of the 4.1 series.

### **4.3.3 JAVASCRIPT**

JavaScript is an object-oriented scripting language used to enable programmatic access to objects with both the client application and other applications. It is primarily used in the form of client-side JavaScript, implemented as an integrated component of the web browser, allowing the development of enhanced user interface and dynamic website. JavaScript is a dialect of the ECMA Script was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

JavaScript, despite the name, is essentially unrelated to the Java programming language even though the two do have superficial similarities. Both languages use syntaxes influenced by that of C syntax, and JavaScript copies many Java names and naming conventions. The language's name is the result of a co-marketing deal between Netscape and sun, in exchange for Netscape bundling sun's Java runtime with their then-dominant browser. The key design principles within JavaScript are inherited from the self and Scheme programming languages. JavaScript™ is a trademark of Sun Microsystems. It was used under license for technology invented and implemented by Netscape communications and current entities such as the Mozilla Foundation.

### **4.3.4 WAMP**

WAMP are packages of independently-created programs installed on computers that use a Microsoft Windows operating system. The interaction of these programs enables dynamic web pages to be served over a computer network, such as the internet or a private network. "WAMP" is an acronym formed from the initials of the operating system (windows) and the package's principal components: Apache, MySQL and PHP(or Perl or python). Apache is a web server, which allows people with web browsers like Internet Explorer or Firefox to

connect to a computer and see information there as web pages. MySQL is a database manager that keeps track of data in a highly organized way. PHP is a scripting language which can manipulate information held in a database and generate web pages afresh each time an element of content is requested from a browser. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

### **4.3.5 ADOBE DREAMWEAVER**

Adobe Dreamweaver is a web development application originally created by Macromedia, and is now developed by Adobe Systems, which acquired Macromedia in 2005. Dreamweaver is available for the both Mac and Windows operating system. Recent versions have incorporated support for web technologies such as CSS, Java Script, and various server side scripting languages and frameworks including ASP, ColdFusion, and PHP.

Dreamweaver allows users to preview websites in locally-installed web servers. It also has site management tools such as FTP/SFTP and WebDAV file transfer and synchronization features, the ability to find and replace lines of text or code by search terms and regular expressions across the entire site, and the tempting feature that allows single source updates of shared code and layout across entire sites without server-side includes or scripting. The behaviours panel also enables use of basic Java Script without any coding knowledge, and integration with Adobe's Spry AJAX frameworks offers easy access to dynamically – generated content and interfaces. Dreamweaver can use third-party “Extensions” to extend core functionality of the application, which any web developer can write (largely in HTML and Java Script). Dreamweaver is supported by a large community of extensions developers who make extensions available (both commercial and free) for most web development tasks from simple rollover effects to full-featured shopping carts. Dreamweaver, like other HTML editors, edits files locally then uploads them to the remote web server using FTP, SFTP, and WebDAV. Dreamweaver CS4 now supports the Subversions (SVN) version control system

## **4.4 System Configuration**

### **4.4.1 Hardware Configuration**

- **Processor** : Intel Pentium and above
- **Display** : VGA Colour Monitor
- **Primary Memory** : 512MB RAM



- **Storage** : 20 GB hard disk
- **Key Board** : Windows 4compatible
- **Mouse** : Windows compatible

#### **4.4.2 Software Configuration**

- **Operating System** : Windows XP or later
- **Front End** : PHP
- **Back End** : MYSQL
- **Server** : WAMP
- **IDE** : Adobe Dreamweaver

## **5. SYSTEM DESIGN AND DEVELOPMENT**

The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The design phase is a transition from a user oriented to a document oriented to the programmers or database personal.

The data flow diagram shows the logical flow of system and defines the boundaries of the system. For a candidate system, it describes the inputs (source), outputs (destination database (files) and procedures (data flow), all in a format that meet the user's requirement. In logical design we specify the user's needs at a level of detail that virtually determines the information flow into out of the system and the required data sources. Following logical design is physical design. This procedures the working system by defining the design specifications that tell programmers exactly what the candidate system must do. In turn we write the necessary programs or modify the software packages that accept input from the user.

### **5.1 USERS OF THE SYSTEM**

- **Admin**
- **Buyer**
- **Seller**
- **Public user**

### **5.2. INPUT DESIGN**

The major objective of the input design is to make the data entry easier, logical and error free. With this objective the screen for the system are developed. The input design requirement such user friendliness, consistent format and interactive dialogue boxes for giving the right message and help for the user at the right time are also considered for the development of the project. The data entry operator need to know the space allocated for each field, the field sequence, which must match with source document and the format in which the data is entered. Thus the objectives of input design are as follows.

Users (Buyer and Seller) Registration Form, The registration form includes the fields such as Name, Phone, Email, Address, Mobile no., profile photo and Adhar number etc. All the fields must be filled and clicks the submit button.

### **5.2.1 Login Form**

The login form includes the fields such as Username and Password. Both the fields must be filled and clicks the login button. If the user enters for the first time, click on the signup button to register.

## **5.3 OUTPUT DESIGN**

Output design should improve the relationship of the system with user and help indecision making. The objective of the output design is to define the format of all printed documents and of the screens that will be produced by the system. The output has been designed as per the needs of the institution. The suggestions of the users are also taken into consideration while designing the layouts and the fields that are to be included in the project.

Operations are required primarily to communicate about the results of the processing to the users and to the management. They also provide the hardcopy of these results for later consultation and rectification. In this system operations are of different form like messages, input boxes and reports. All recent reports can be generated by the system. Menu will help the user to select the required outputs. The details in the reports must be clear, descriptive and simple to understand to the user.

## **5.4 DATABASE DESIGN**

Database design is one of the most important parts of the system design phase. In a database environment common data are available and are used by several users. Instead of each program managing its own data, authorized users share data across application with the database software managing the data as an entity. The primary objective of a database design are fast response time to enquiries, more information at low cost, control of redundancy, clarity and ease of use, date and program independence, accuracy and integrity of the system, fast recovery and availability of powerful end-user languages. The theme behind a database is to handle information as an integrated whole thus the main objective is to make information as access easy, quick, inexpensive and flexible for the users.

Data directory specifies the major element in the system, and care should be taken while designing, in order to avoid unnecessary duplication of data. The entire package depends on how the data's are maintained in the system. Several tables are maintained in the system

to store data that are required for the processing of various data as well as storing intermediate or final processed results.

Database design mainly aims at handling large volumes of information, involving the definitions for the structure of storage and provisions for the manipulation of information, providing safety of information despite of system crashes due to unauthorized access. Some conditions are satisfied in database design stage.

- Control redundancy
- Ease of use
- Data independency
- Accuracy and integrity
- Recovery from failures
- Security and privacy
- Performance

### **5.4.1 Normalization**

Normalization is the process of decomposing a set of relations with anomalies to produce smaller and well-structured relations that contain minimum redundancy. It is a formal process of deciding which attributes should be grouped together in a relation. The purpose of normalization is to make table as simple as possible.

**Normalization carried out in the system for following reasons:**

- To structure the data. So that there is no repetition of data, this helps in saving space.
- To permit simple retrieval of data in response to query and request.
- To simplify maintenance of data through updates, insertion and deletion.
- To reduce the need to restructures or recognize data which new application requirement arise.
- Primary key is assigned for this purpose. The primary key fields in almost all tables helps to ease the search and improve efficiency

#### **5.4.1.1 First Normal Forms**

First Normal form (1NF) is now considered to be part of the formal definition of relational model. 1NF is designed to disallow multi-valued attribute, composite attributes, and their combinations. It states that the domain of an attribute must include only atomic values. A domain is atomic, if elements of the domains are considered to be indivisible units. We say

that a relational schema R is in 1NF if the domain of all attributes of 'R' is atomic. The tables included in developing the user register contain atomic values and hence it is in 1NF.

#### **5.4.1.2 Second Normal Form**

Second Normal form (2NF) is based on the concept of functional dependency. A relation R is in 2NF if it is in 1NF and every non key attribute A of R is fully dependent on the primary key.

That is, relation is said to be in 2NF if each attribute A in R meets one of the following criteria:

- 1. It appears in the primary key.**
- 2. It is fully functionally dependent on the primary key.**

The tables designed in the proposed system, contain a primary key for uniquely identifying each user.

#### **5.4.1.3 Third Normal Form**

Third Normal form (3NF) is based on the concept of transitive dependency. A relation is said to be in 3NF if it is in 2NF and has no transitive dependencies. That is all the non key attribute should be functionally determined by the primary key. In the proposed system all attributes of tables are fully depends on the primary key only that is all non-key attributes are mutually independent. We use third normal form in our project

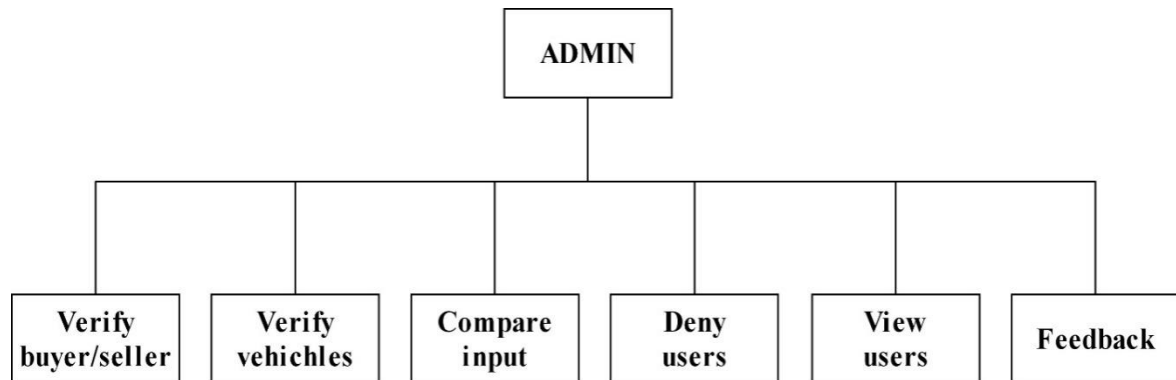
### **5.5 Web Page Design**

In computer software development the web page design face has a important part role. The pages are used to gather user inputs and to display the information to the user. An excellent design of pages will improve the quality of application. It will help to increase the probability of user acceptance. Efficient page design can reduces the data entry time and it helps the users to see the different controls placed in the pages. A good page design simplifies the data access and entry.

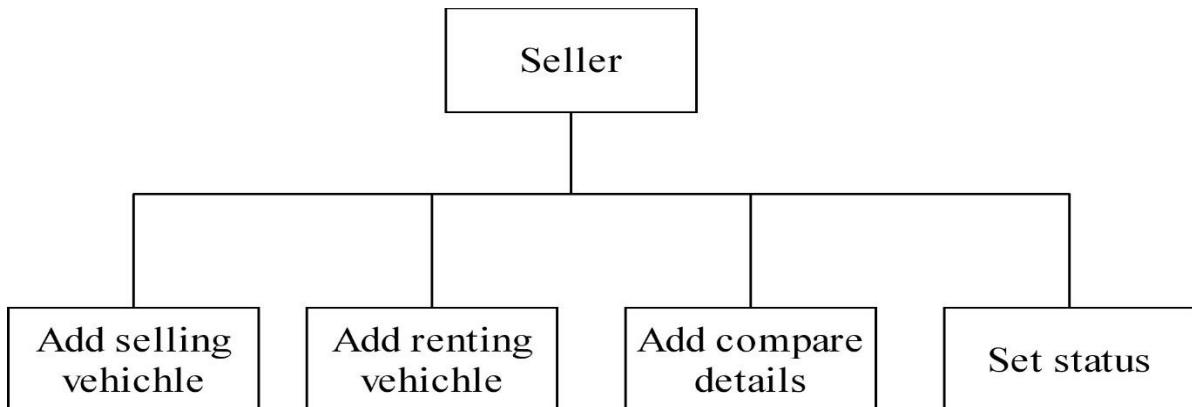
In this project the pages are designed in a logical order. The controls are placed based on their property. The tab order is set in sequential order. The user needs to press the tab key to jump to next field with pressing enter key. The parts which need more importance are highlighted with colours. It helps the user to detect these portions of pages.

## 5.6 PROCESS DESIGN

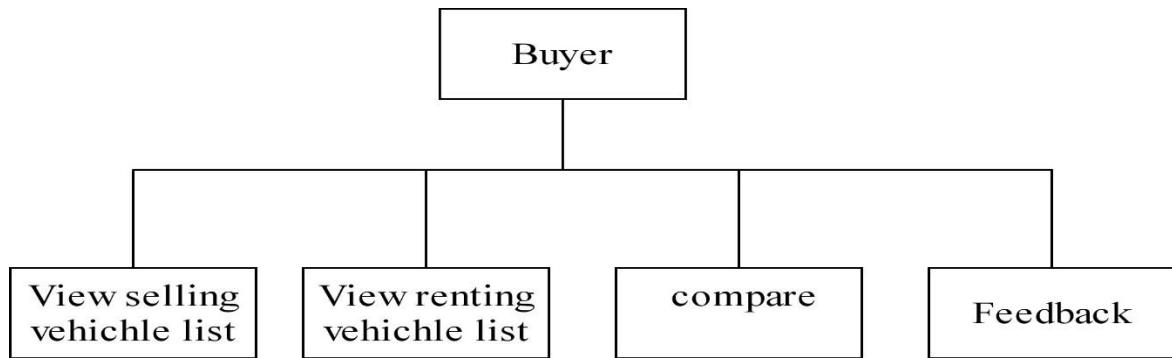
### 5.6.1 Admin



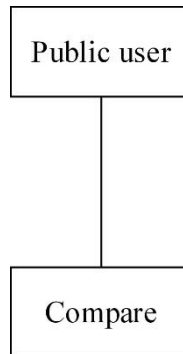
### 5.6.2 Seller



### 5.6.3 Buyer



### 5.6.4 Public user



## **6 SYSTEM IMPLEMENTATION & TESTING**

### **6.1 SYSTEM IMPLEMENTATION**

Implementation of Parabello system is very cost effective and very simple. Finding good quality vehicles and authorized sellers is very tedious task in online marketplace. The timely availability of desired car is now practically impossible. Even most of the buyers and sellers are still in trouble for selling or buying a car. The project “Parabello System” is a solution for major pitfalls of existing scenario.

Here the system manages all types of selling and buying with proper management of users and authenticating them. The users can find sellers matching their expectation. Also they can negotiate the financial.

The system facilitates the seller or buyer to register independently. The profiles of sellers are available for comparing between two desired cars, gives a best opportunity to users to select a car with good feedback and better performance. The project is also deals with rental service with verified renters.

### **6.2 SYSTEM TESTING**

#### **6.2.1 TYPES OF TESTING**

Testing is very important in determining the reliability and efficiency of software, and hence it is very crucial stage in software development. Tests are conducted on the software to evaluate its performance at different levels.

Software testing determines the correctness, completeness, and quality of software being developed. Validation refers to the process of checking that the developed software meets the requirements specified by the user. The activities involved in the testing phase basically evaluate the capability of that system meets its requirements. The main objective of software testing is to detect errors in the software. Errors occur if some part of the developed system is found to be incorrect, incomplete or inconsistent. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects). It involves the execution of a software component or system to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- Meets the requirements that guided its design and development



- Responds correctly to all kinds of inputs
- Performs its functions within an acceptable time is sufficiently usable
- can be installed and run in its intended environments
- Achieves the general result its stake holders desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects). Software testing can provide objective, independent information about the quality of software and risk of its failure to users and sponsors. Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted.

For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an Agile approach, requirements, programming, and testing are often done concurrently.

#### **6.2.1.1 White-box Testing**

Tests are performed to ensure that all internal operations of the software are performed according to the specifications of the client. This is called White box testing. White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

The details entered by the administrator are saved and stored in the database, and testing is done to verify whether the control of each form or action is working in the exact way.

#### **6.2.1.2 Black-box Testing**

Tests are performed to ensure that each function is working properly. This is referred to as Black box testing. Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output without any knowledge of the test object's internal structure. Testing is conducted in the system so that the functions namely Login, sending requests, searching a particular cattle, or a particular record etc. are done properly

### **6.3 TESTING STRATEGY**

#### **6.3.1 Condition Testing**

Test cases are derived to determine whether the logic conditions and decision statements are free from errors. Condition testing strategy is used to check if the operators used are correct and to verify conditions such as if an error message is displayed if a non-approved seller wants to login to the system, or a seller wants to register without giving his email or any other mandatory fields from the seller register table.

#### **6.3.2 Loop Testing**

This testing is used to check the variety of loops present in programming. The working of the loops such as while, for and do while are checked for its proper execution. The statements inside the loop body are executed line by line for every condition that satisfies the loop.

#### **6.3.3 Unit Testing**

The process on each individual module is ensured that it functions properly as a unit. Sample data is given for unit testing. The unit test results are recorded for further references. During unit testing the functionality of the program unit, validation and limitation are tested. Unit test makes every user white boxing technique, exercising

specific paths in a modular box control structure to ensure complete coverage maximum error detection.

#### **6.3.4 Integration Testing**

The modules are integrated to form complete software package. It addresses the issues associated with given problem of verification and program construction. Test that part of the system at some level work together correctly.

#### **6.3.5 System Testing**

After performing the integration testing, the next step is output testing of the proposed system.

No system could be useful if it doesn't produce the required output in a specified format. The output generated are displayed by the system under consideration and then tested by comparing with the format require by the user. Here the output format is considered into two ways, one in on-screen and other in printed format.

#### **6.3.6 Validation Testing**

Validation testing provides the final assurance that software meets all functions, behavioural and performance requirements.

#### **6.3.7 User Acceptance Testing**

Usability testing is a means for measuring how well people can use some human-made object (such as a web page, a computer interface a document, or a device) for its intended purpose, i.e. usability testing measures the usability of the object . During usability testing, the aim is to observe people using the product in as realistic a situation as possible, to discover errors and areas of improvement. Usability testing usually involves a controlled experiment to determine how well user can use the product.

## 7. SYSTEM SECURITY MEASURES

System security is a branch of technology known as information security as applied to computers and networks. The objective of system security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The terms system security, means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The technologies of system security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behaviour.

### 7.1 Checks and Controls

Some of the important checks and controls used for system security are given below, they are:

**Confidentiality:** A security measure which protects against the disclosure of information to parties other than the intended users that is by no means the only way of ensuring.

**Integrity:** A measure intended to allow the receiver to determine that the information which it receives has not been altered in transit or by other than the originator of the information. Integrity schemes often use some of the same underlying technologies as confidentiality schemes, but they usually involve adding additional information to a communication to form the basis of an algorithmic check rather than the encoding all of the communication.

**Authentication:** A measure designed to establish the validity of a transmission, message, or originator. Allows a receiver to have confidence that information it receives originated from a specific known source.

**Authorization:** The process of determining that a requester is allowed to receive a service or perform an operation. Access control is an example of authorization.

**Availability:** Assuring information and communications services will be ready for use when expected. Information must be kept available to authorized persons when they need it.

**Non-repudiation:** A measure intended to prevent the later denial that an action happened, or a communication that took place etc. In communication terms this often involves the interchange of authentication information combined with some form of provable time stamp.

## 7.2 Database Security

Data security is the practice of keeping data protected from corruption and unauthorized access. The focus behind data security is to ensure privacy while protecting personal or corporate data. Data is the raw form of information stored as columns and rows in our databases, network servers and personal computers. This may be a wide range of information from personal files and intellectual property to market analytics and details intended to top secret. Data could be anything of interest that can be read or otherwise interpreted in human form.

Encryption has become a critical security feature for thriving networks and active home users alike. This security mechanism uses mathematical schemes and algorithms to scramble data into unreadable text. It can only be decoded or decrypted by the party that possesses the associated key.

Full-disk encryption offers some of the best protection available. This technology enables you to encrypt every piece of data on a disk or hard disk drive. Full disk encryption is even more powerful when hardware solutions are used in conjunction with software components. This combination is often referred to as end-based or end-point full disk.

Authentication is another part of data security that we encounter with everyday computer usage. Just think about when you log into your email or blog account. That single sign-on process is a form authentication that allows you to log into applications, files, folders and even an entire computer system. Once logged in, you have various given privileges until logging out. Some systems will cancel a session if your machine has been idle for a certain amount of time, requiring that you prove authentication once again to re-enter. The single sign-on scheme is also implemented into strong user authentication systems. However, it requires individuals to login using multiple factors of authentication.

Data security wouldn't be complete without a solution to backup your critical information. Though it may appear secure while confined away in a machine, there is always a chance that your data can be compromised. You could suddenly be hit with a malware infection where a virus destroys all of your files. Someone could enter your computer and thief data by sliding through a security hole in the operating system. Perhaps it was an inside job

that caused your business to lose those sensitive reports. If all else fails, a reliable backup solution will allow you to restore your data instead of starting completely from scratch.

### **7.3 Creation of User Profiles and Access Rights**

User security lets your application use security rules to determine what it displays. It has two elements:

#### **Authentication:**

Authentication is another part of user security that we encounter with everyday computer usage. Ensures that a valid user is logged-in, based on an ID and password provided by the user. ColdFusion (or, in some cases if you use web server authentication, the web server) maintains the user ID information while the user is logged-in.

#### **Authorization:**

Ensures that the logged-in user is allowed to use a page or perform an operation. Authorization is typically based on one or more *roles* (sometimes called groups) to which the user belongs. For example ,when the user signing as a buyer he/she can only see the property details for selling and also search for properties. But they cannot modify the property details. You can also use the user ID for authorization.

## **8. CONCLUSION**

The system “PARABELLO” has been developed for the given conditions and is found working effectively under all circumstances that may arise in the real environment. The screens designs have been improved to a great extend so as to make the user comfortable with the system. The system was tested for a wide range of input and found to be error-free in all test cases. Even a user with no knowledge of the technical aspects of the system can easily operate it.

The System is highly user friendly, interactive and operations are reliable. It provides security since authentication is needed by the users of some applications. The system is highly flexible among different client applications. On the whole the project is technically feasible and viable.

### **Finding**

The project has been completed successfully with the maximum satisfaction of the organization. All possible measures are taken to ensure that the accuracy of the output is maintained. The system has been tested with the real data and the results are found to be satisfactory. The project has been left for chance of future enhancement to meet the future needs and technologies.

- It offers a user-friendly website
- Inexpensive worldwide distribution
- Easy content updating

### **Limitations**

The major limitation is the Availability of this system. Though other manual systems are available everywhere. This system is available anywhere in the world only if there is the Internet facility.

The major limitations are

- Less availability of system
- Need of internet facility
- Waiting for admin response is time consuming

## **9. SCOPE FOR FUTURE WORKS**

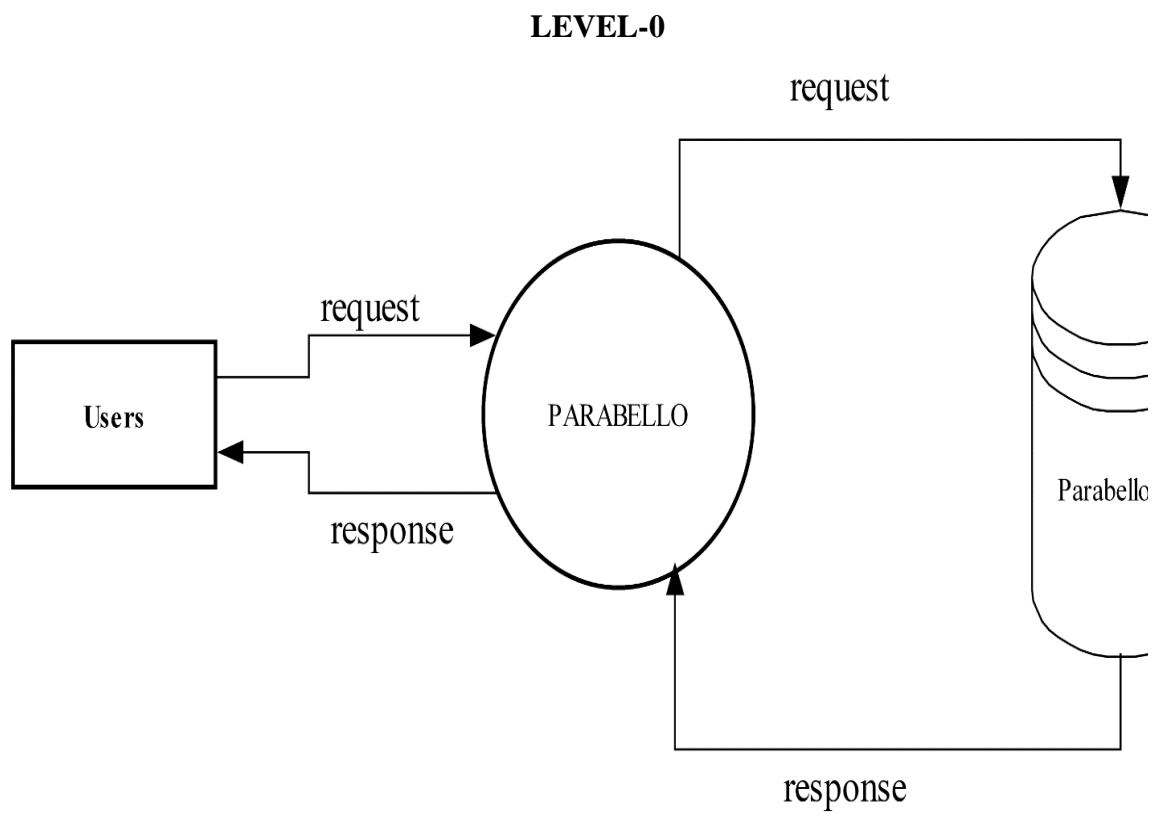
The software is developed in PHP which makes the system more reliable and compatible with the other environments. The application proves better extensibility and flexibility for future enhancements. Any further requirement application is possible with the same features guaranteed. The design of this software is in such a way that the addition of any new module if necessary is possible without affecting the integrity of the present system.

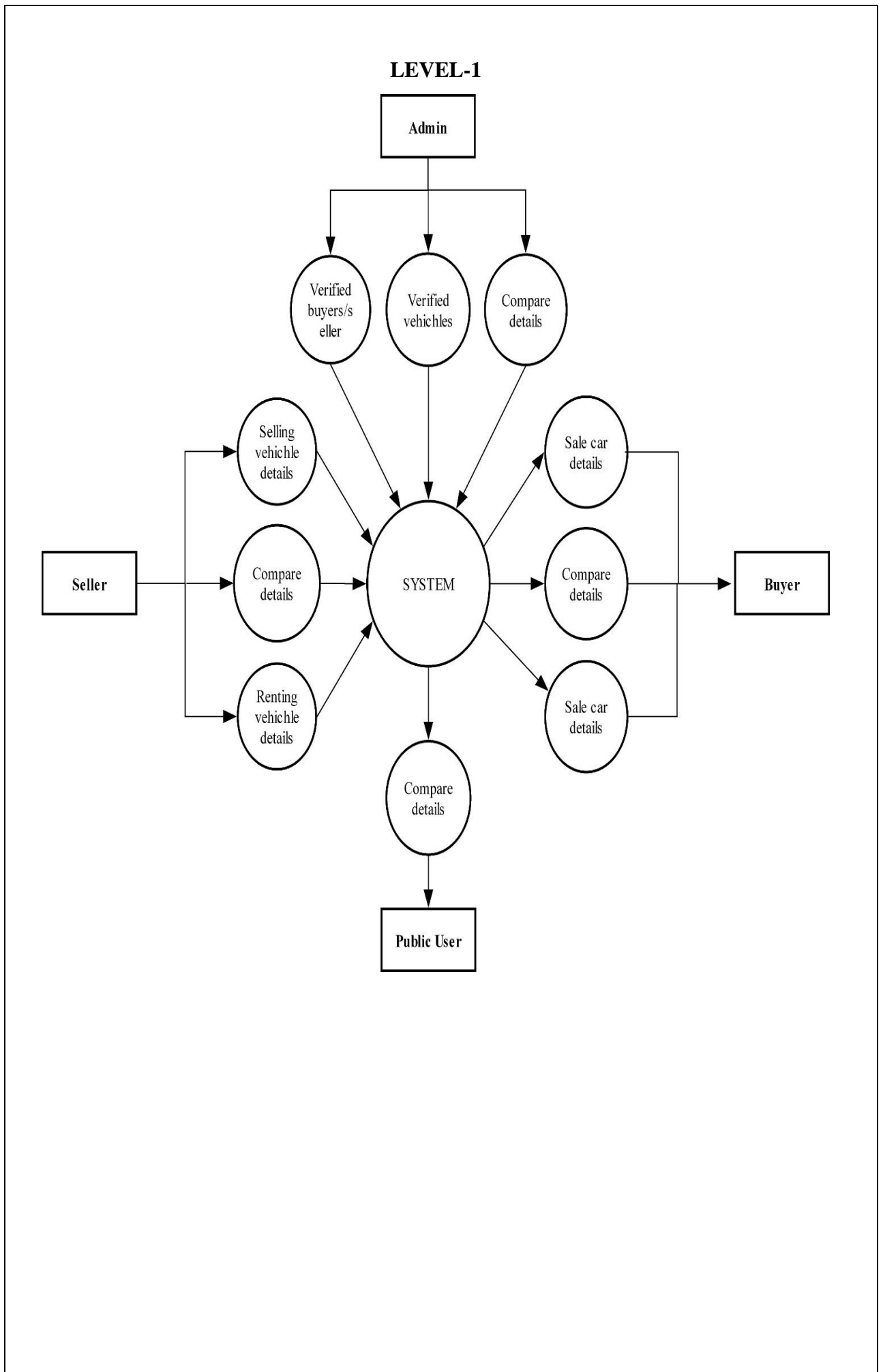
The proposed site considers mainly on giving information about online sellers of Parabello system and other facilities to the users more accurately and efficiently. And future enhancement is possible to include many more features like add the option for pasting photos of the user and make a mobile alert when a user get an email from admin. For further enhancement we can develop this website as a all India based website. Design of the system has done foreseeing future enhancements so it will not affect the base system inversely.



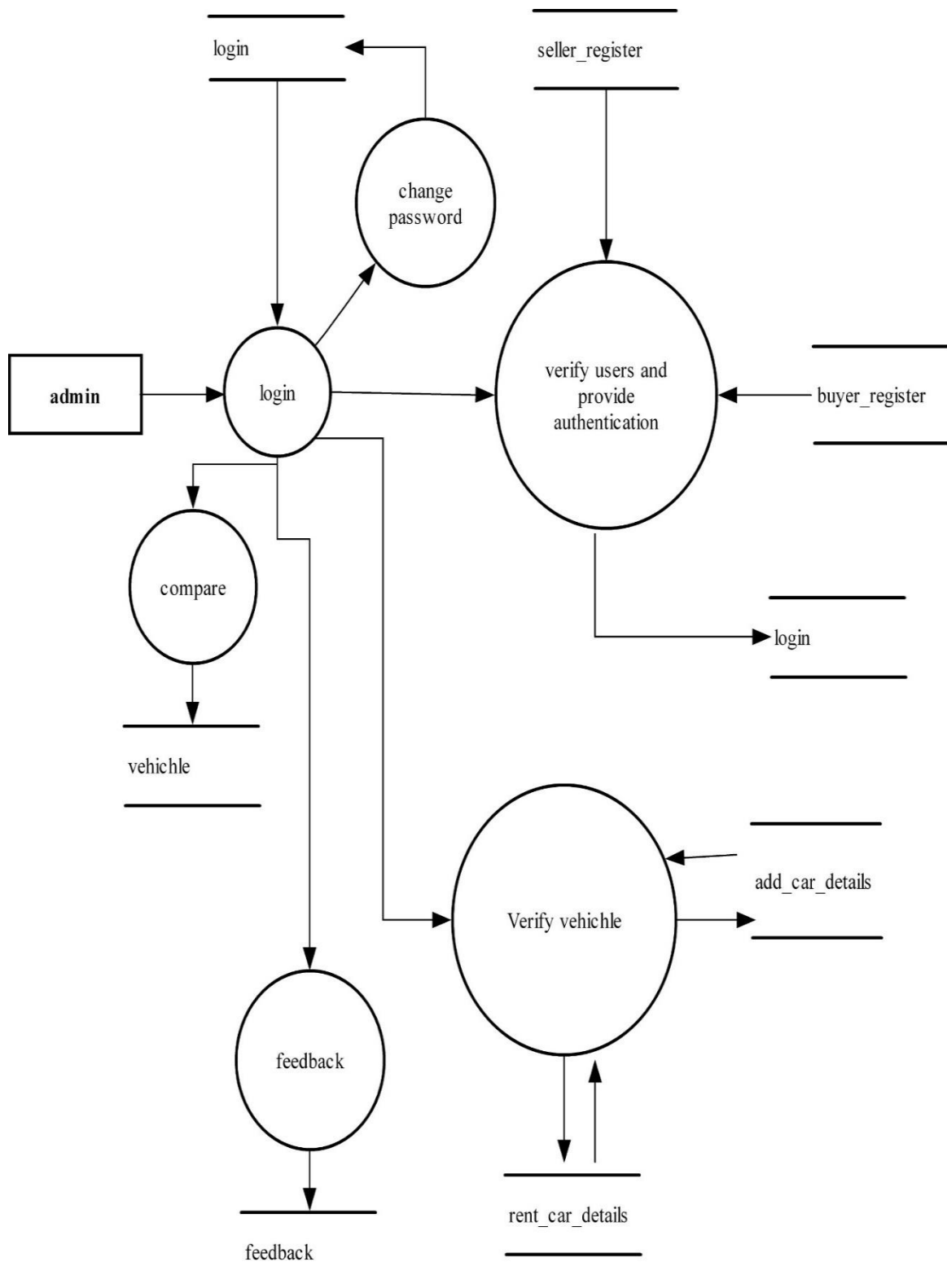
## 10. APPENDIX

### 10.1 DATA FLOW DIAGRAM (DFD)

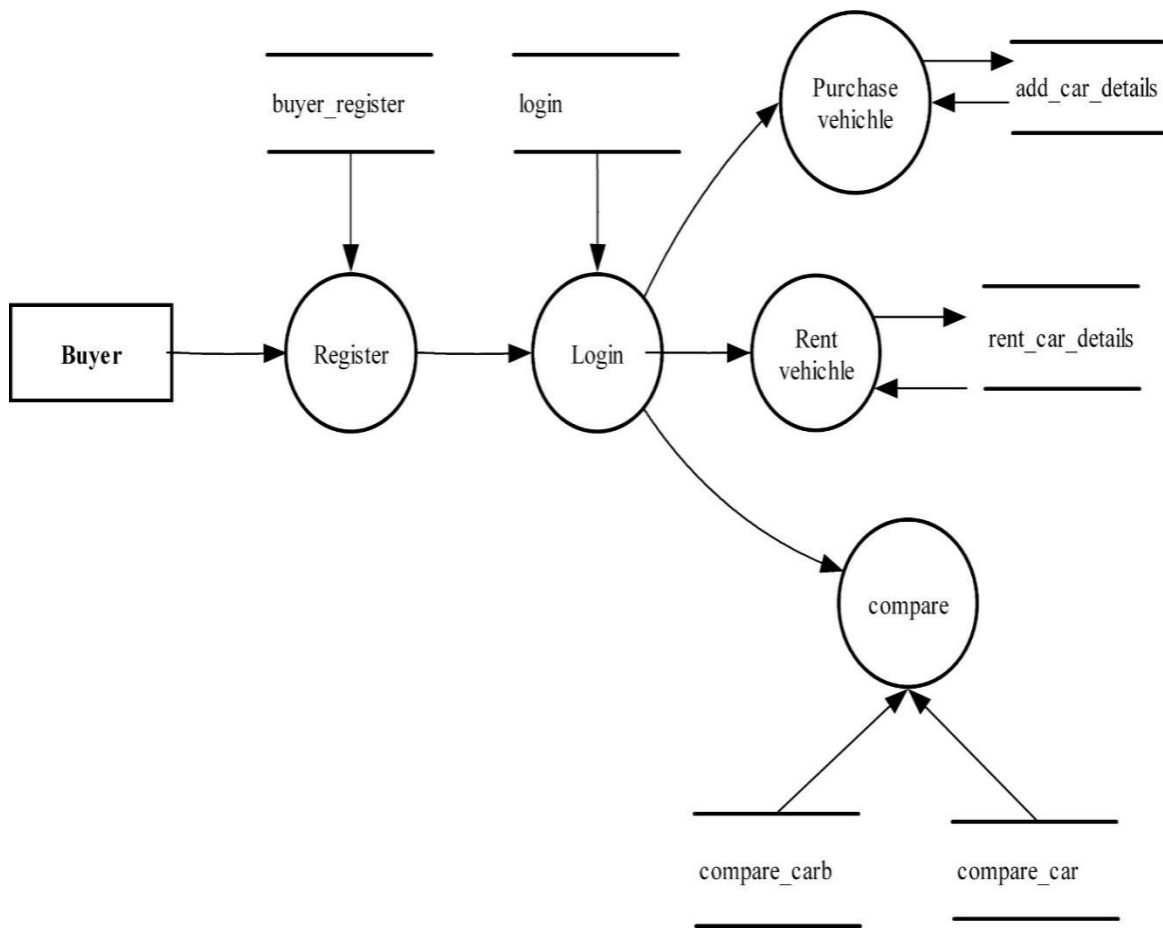




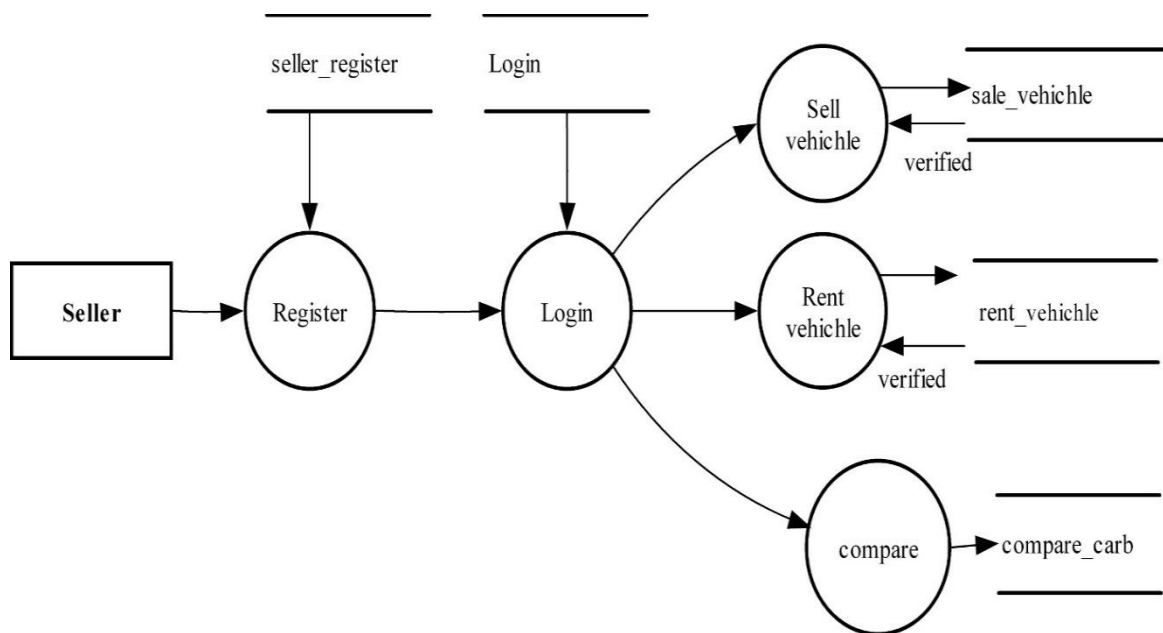
## LEVEL-2.1



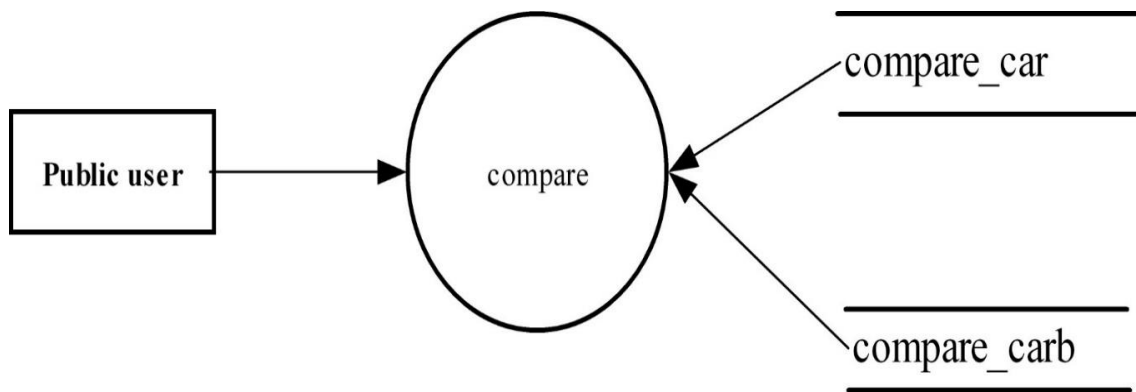
### LEVEL-2.2



### LEVEL-2.3



## LEVEL-2.4



### 10.2 Tables structure

#### 1. Table Name: adminlogin

| No | Column name | Data type   | Constraints           |
|----|-------------|-------------|-----------------------|
| 1  | username    | varchar(45) | Primary key, Not null |
| 2  | password    | varchar(45) | Not null              |

#### 2. Table Name: feedback

| No | Column name | Data type   | Constraints         |
|----|-------------|-------------|---------------------|
| 1  | Id          | int         | Primarykey, Notnull |
| 2  | Name        | varchar(35) | Not null            |
| 3  | comment     | text        | Not null            |

**3. Table Name:** add\_car\_details

| No | Column name | Data type    | Constraints              |
|----|-------------|--------------|--------------------------|
| 1  | no          | int          | Primary key,<br>Not null |
| 2  | company     | varchar(35)  | Not null                 |
| 3  | model       | int          | Not null                 |
| 4  | color       | varchar(30)  | Not null                 |
| 5  | kms_driven  | int          | Not null                 |
| 6  | fuel_type   | varchar(25)  | Not null                 |
| 7  | image       | varchar(100) | Not null                 |
| 8  | Image2      | varchar(100) | Not null                 |
| 9  | Image3      | varchar(100) | Not null                 |
| 10 | Image4      | varchar(100) | Not null                 |
| 11 | city        | varchar(50)  | Not null                 |
| 12 | rate        | int          | Not null                 |
| 13 | status      | varchar(20)  | Not null                 |
| 14 | priority    | int          | Not null                 |
| 15 | seller_no   | int          | Foreign key,<br>Not null |

**4. Table Name:** buyer\_register

| No | Column name | Data type    | Constraints           |
|----|-------------|--------------|-----------------------|
| 1  | no          | int          | Primary key, Not null |
| 2  | name        | varchar(50)  | Not null              |
| 3  | mobile_no   | int          | Not null              |
| 4  | adhar_no    | int          | Not null              |
| 5  | email       | varchar(60)  | Not null              |
| 6  | city        | Varchar(50)  | Not null              |
| 7  | address     | text         | Not null              |
| 8  | status      | int          | Not null              |
| 9  | photo       | varchar(100) | Not null              |

**5. Table Name:** compare\_car

| No | Column name       | Data type    | Constraints           |
|----|-------------------|--------------|-----------------------|
| 1  | No                | int          | Primary key, Not null |
| 2  | car_name          | varchar(50)  | Not null              |
| 3  | Yom               | int          | Not null              |
| 4  | Mileage           | int          | Not null              |
| 5  | seat_capacity     | int          | Not null              |
| 6  | transmission_type | varchar(50)  | Not null              |
| 7  | Safety            | varchar(150) | Not null              |
| 8  | Price             | int          | Not null              |

|    |                     |              |          |
|----|---------------------|--------------|----------|
| 9  | body_type           | varchar(150) | Not null |
| 10 | comfort_convinience | varchar(60)  | Not null |
| 11 | maitenance_cost     | int          | Not null |

**6.Table Name:** compare\_carb

| No | Column name         | Data type    | Constraints          |
|----|---------------------|--------------|----------------------|
| 1  | no                  | int          | Primary key,Not null |
| 2  | car_name            | varchar(50)  | Not null             |
| 3  | yom                 | int          | Not null             |
| 4  | mileage             | int          | Not null             |
| 5  | seat_capacity       | int          | Not null             |
| 6  | transmission_type   | varchar(50)  | Not null             |
| 7  | safety              | varchar(150) | Not null             |
| 8  | price               | int          | Not null             |
| 9  | body_type           | varchar(150) | Not null             |
| 10 | comfort_convinience | varchar(60)  | Not null             |
| 11 | maitenance_cost     | int          | Not null             |



**7. Table Name:** login

| No | Column name | Data type   | Constraints          |
|----|-------------|-------------|----------------------|
| 1  | username    | varchar(50) | Primary key,Not null |
| 2  | password    | varchar(60) | Not null             |
| 3  | id          | int         | Foreign key,Not null |
| 4  | role        | varchar(20) | Not null             |

**8. Table Name:** seller\_register

| No | Column name | Data type    | Constraints          |
|----|-------------|--------------|----------------------|
| 1  | no          | int          | Primary key,Not null |
| 2  | name        | varchar(50)  | Not null             |
| 3  | mobile_no   | bigint       | Not null             |
| 4  | email       | varchar(60)  | Not null             |
| 5  | city        | Varchar(50)  | Not null             |
| 6  | address     | text         | Not null             |
| 7  | status      | int          | Not null             |
| 8  | photo       | varchar(100) | Not null             |

**9. Table Name:**rent\_car\_details

| No | Column name | Data type    | Constraints          |
|----|-------------|--------------|----------------------|
| 1  | no          | int          | Primary key,Not null |
| 2  | company     | varchar(35)  | Not null             |
| 3  | model       | int          | Not null             |
| 4  | color       | varchar(30)  | Not null             |
| 5  | kms_driven  | int          | Not null             |
| 6  | fuel_type   | varchar(25)  | Not null             |
| 7  | image       | varchar(100) | Not null             |
| 8  | Image2      | varchar(100) | Not null             |
| 9  | Image3      | varchar(100) | Not null             |
| 10 | Image4      | varchar(100) | Not null             |
| 11 | city        | varchar(50)  | Not null             |
| 12 | rate        | int          | Not null             |
| 13 | status      | varchar(20)  | Not null             |
| 14 | priority    | int          | Not null             |
| 15 | seller_no   | int          | Foreign key,Not null |

## 11. BIBLIOGRHY

### 11.1 BOOKS:

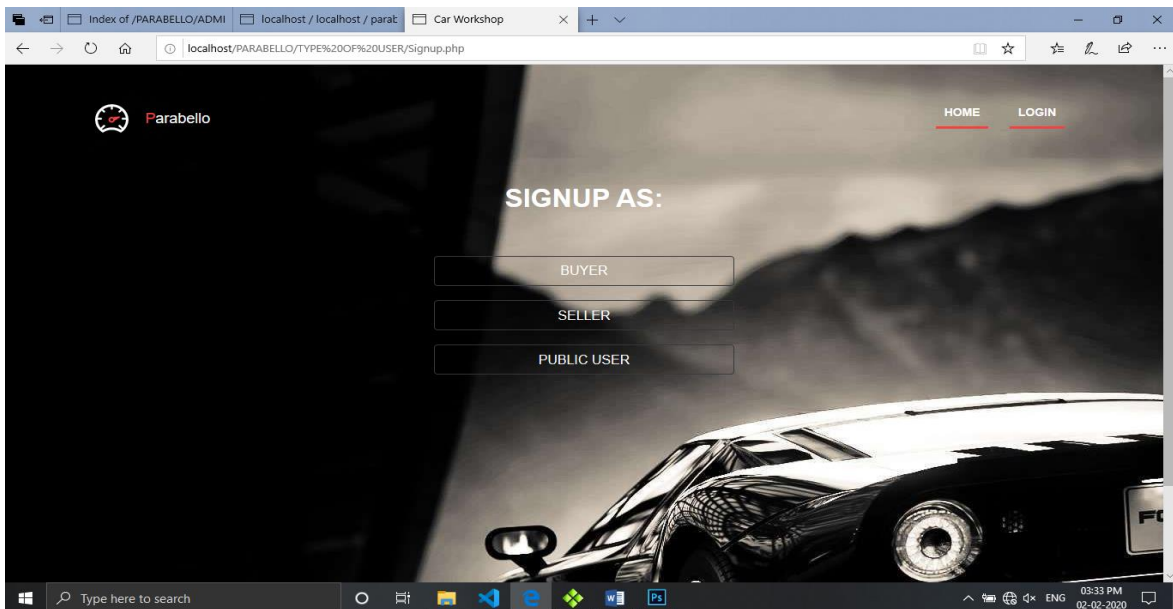
1. Database system concept- Marvin .F.Korth 2.
- System Analysis And design- Marvin Gore

### 11.2 WEBSITES:

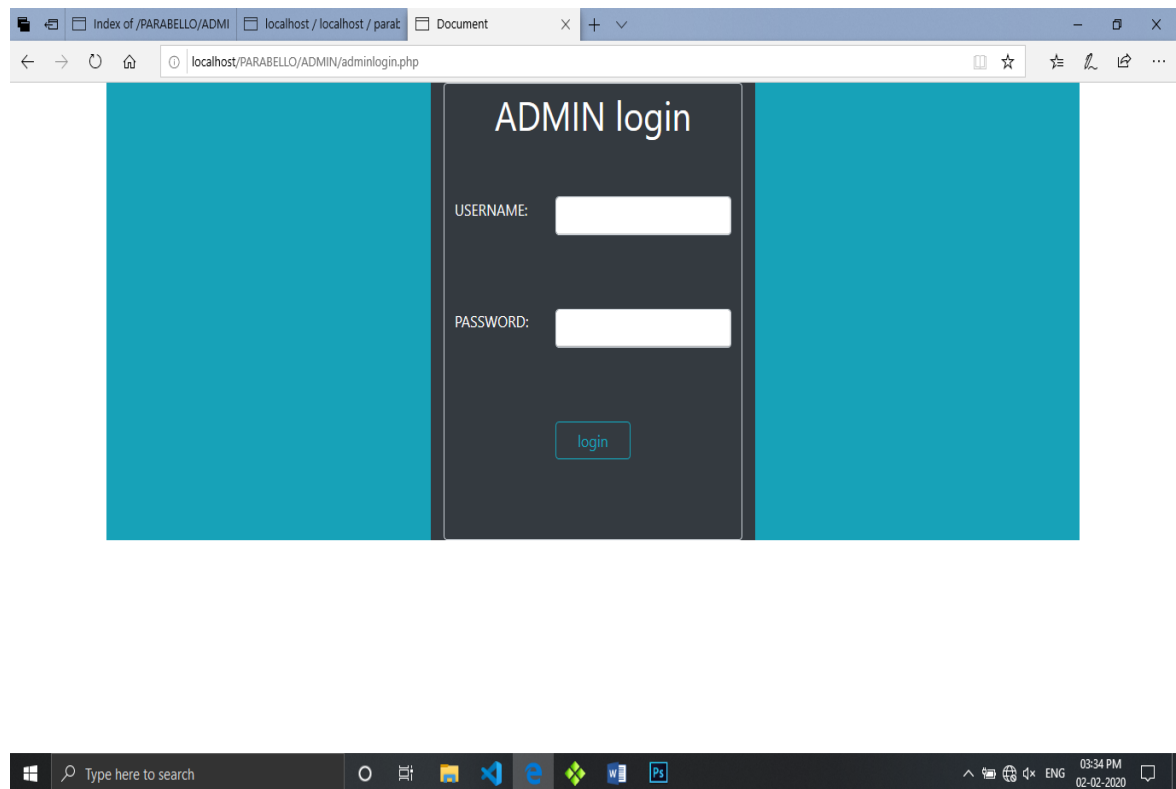
1. www.google.com
2. ww.W3school.com

## 12 SAMPLE SCREEN FORMATS

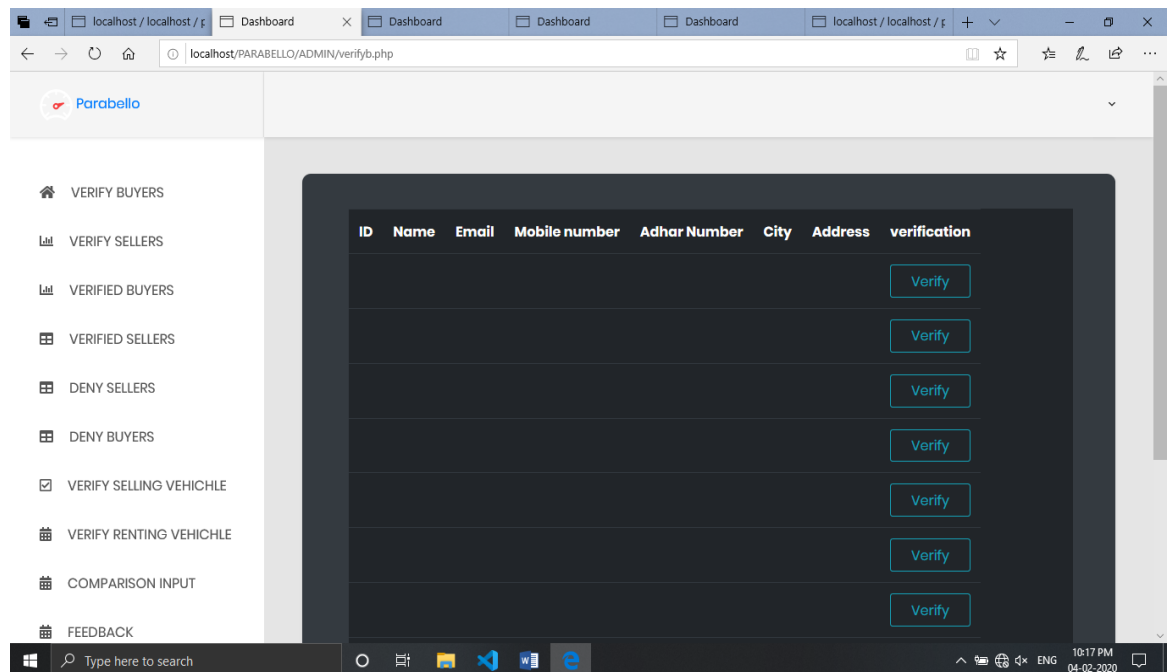
### Homepage



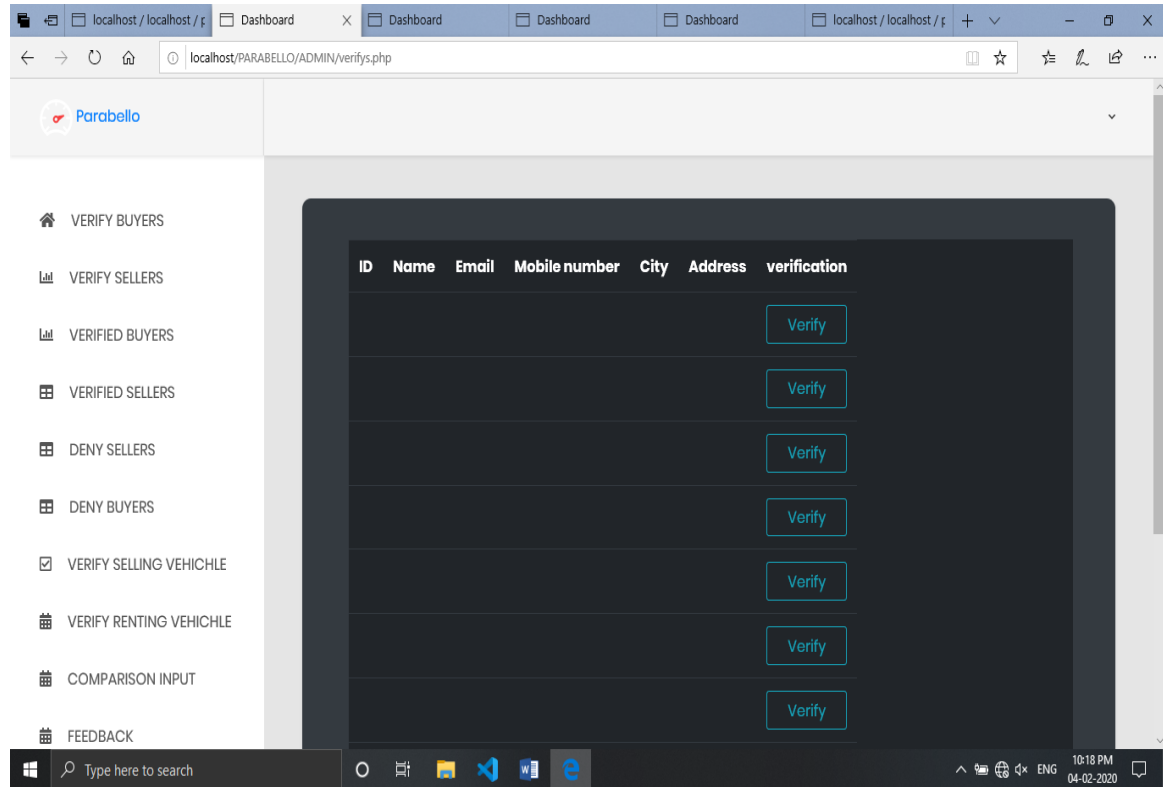
## Admin Login Page



## Verify buyer



## Verify seller

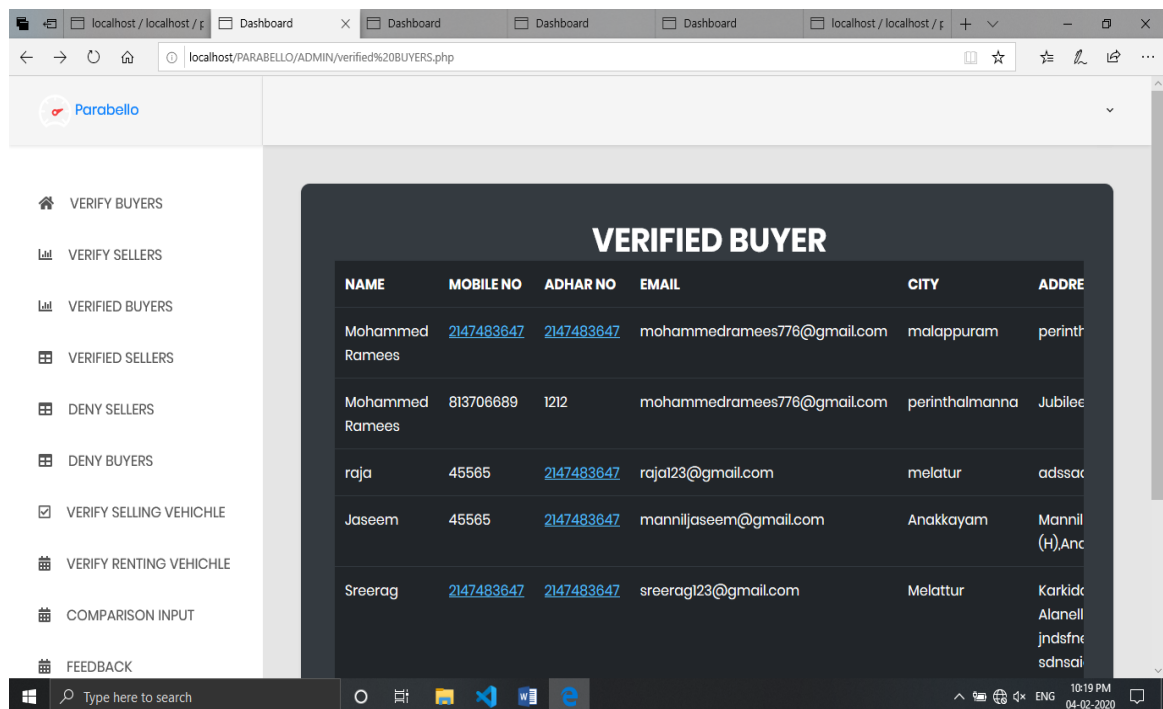


Parabello

- VERIFY BUYERS
- VERIFY SELLERS
- VERIFIED BUYERS
- VERIFIED SELLERS
- DENY SELLERS
- DENY BUYERS
- VERIFY SELLING VEHICLE
- VERIFY RENTING VEHICLE
- COMPARISON INPUT
- FEEDBACK

| ID | Name | Email | Mobile number | City | Address | verification |
|----|------|-------|---------------|------|---------|--------------|
|    |      |       |               |      |         | Verify       |
|    |      |       |               |      |         | Verify       |
|    |      |       |               |      |         | Verify       |
|    |      |       |               |      |         | Verify       |
|    |      |       |               |      |         | Verify       |
|    |      |       |               |      |         | Verify       |

## Verified buyers



Parabello

- VERIFY BUYERS
- VERIFY SELLERS
- VERIFIED BUYERS
- VERIFIED SELLERS
- DENY SELLERS
- DENY BUYERS
- VERIFY SELLING VEHICLE
- VERIFY RENTING VEHICLE
- COMPARISON INPUT
- FEEDBACK

| VERIFIED BUYER  |                            |                            |                             |                |               |
|-----------------|----------------------------|----------------------------|-----------------------------|----------------|---------------|
| NAME            | MOBILE NO                  | ADHAR NO                   | EMAIL                       | CITY           | ADDRESS       |
| Mohammed Ramees | <a href="#">2147483647</a> | <a href="#">2147483647</a> | mohammedramees776@gmail.com | malappuram     | perinth       |
| Mohammed Ramees | 813708689                  | 1212                       | mohammedramees776@gmail.com | perinthalmanna | Jubilee       |
| raja            | 45565                      | <a href="#">2147483647</a> | raja123@gmail.com           | melatur        | adssai        |
| Jaseem          | 45565                      | <a href="#">2147483647</a> | manniljaseem@gmail.com      | Anakkayam      | Mannil (H)Anc |
| Sreerag         | <a href="#">2147483647</a> | <a href="#">2147483647</a> | sreerag123@gmail.com        | Melattur       | Karkid Alane  |

## Verified seller

**VERIFIED SELLERS**

| USERNAME        | EMAIL                       | MOBILE NO                               | CITY       | ADDRESS        |
|-----------------|-----------------------------|---|------------|----------------|
| Mohammed Ramees | mohammedramees776@gmail.com | <a href="tel:2147483647">2147483647</a> | malappuram | perinthalthana |
| mahir           | mahirhussain@gmail.com      | 7012959287                              | MAMPAD     | NILAMBUR       |
| rava            | asd24@gmail.com             | 7012959287                              | pmna       | fsdafdsaf      |
| ramu            | ramu123@gmail.com           | 941616152                               | malappuram |                |
| Mohammed Ramees | mohammedramees776@gmail.com | 1212                                    | malappuram | perinthalthana |
| Mohammed Ramees | mohammedramees776@gmail.com | 1212                                    | malappuram | perinthalthana |

## Deny seller

| Mobile number               | City                                    | Address    | verification   |                       |
|-----------------------------|---|------------|----------------|-----------------------|
| mohammedramees776@gmail.com | <a href="tel:2147483647">2147483647</a> | malappuram | perinthalthana | <button>Deny</button> |
| rhussain@gmail.com          | 7012959287                              | MAMPAD     | NILAMBUR       | <button>Deny</button> |
| 4@gmail.com                 | 7012959287                              | pmna       | fsdafdsaf      | <button>Deny</button> |
| 123@gmail.com               | 941616152                               | malappuram |                | <button>Deny</button> |
| mohammedramees776@gmail.com | 1212                                    | malappuram | perinthalthana | <button>Deny</button> |
| mohammedramees776@gmail.com | 1212                                    | malappuram | perinthalthana | <button>Deny</button> |

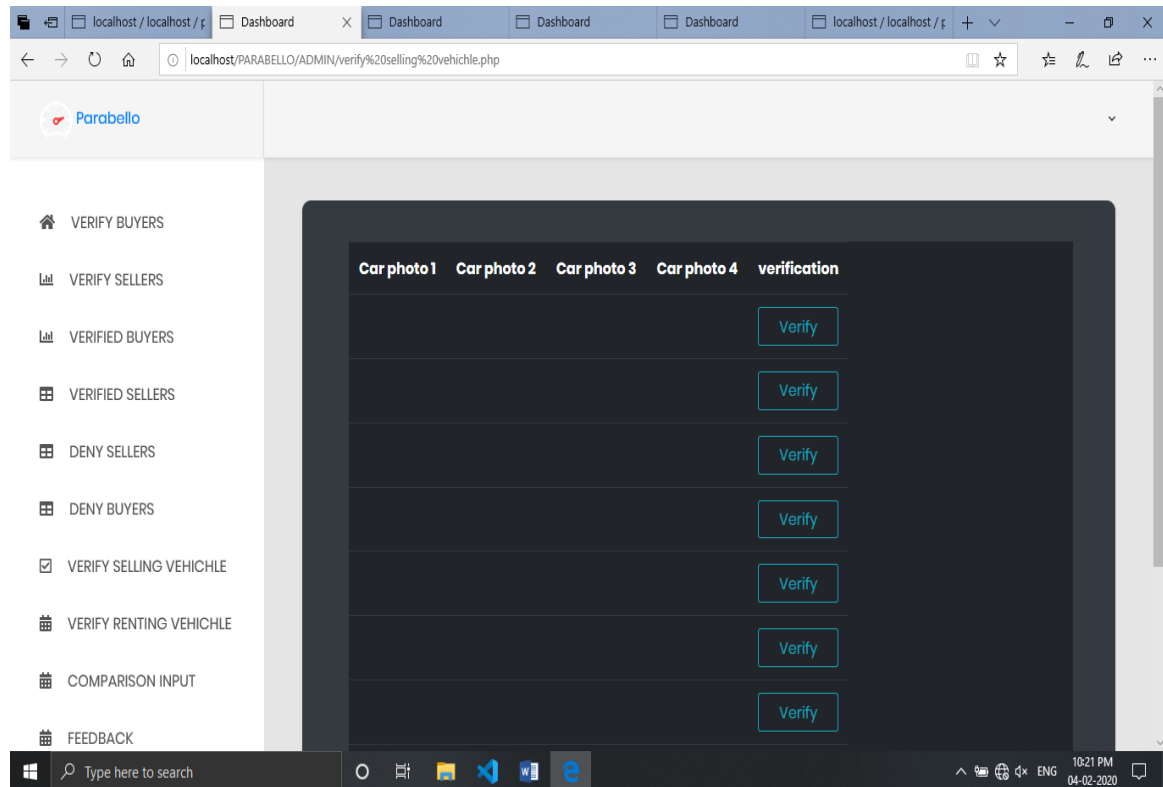
## Deny buyer

| ID | Name            | Email                      | Mobile number              | Adhar Number                | City           |
|----|-----------------|----------------------------|----------------------------|-----------------------------|----------------|
| 4  | Mohammed Ramees | <a href="#">2147483647</a> | <a href="#">2147483647</a> | mohammedramees776@gmail.com | malappuram     |
| 5  | Mohammed Ramees | 813706689                  | 1212                       | mohammedramees776@gmail.com | perinthalmanna |
| 6  | raja            | 45565                      | <a href="#">2147483647</a> | rajal23@gmail.com           | melatur        |
| 7  | Jaseem          | 45565                      | <a href="#">2147483647</a> | manniljaseem@gmail.com      | Anakkayam      |
| 8  | Sreerag         | <a href="#">2147483647</a> | <a href="#">2147483647</a> | sreerag123@gmail.com        | Melattur       |

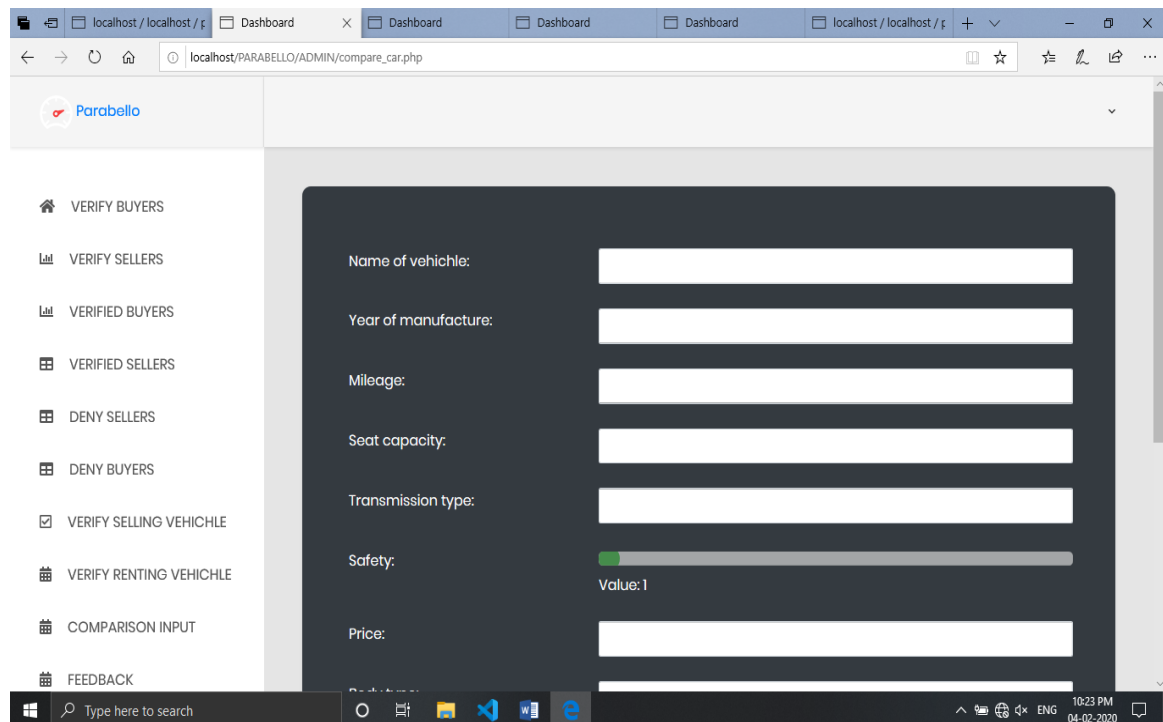
## Verify renting vehicle

| Car photo 1 | Car photo 2 | Car photo 3 | Car photo 4 | verification           |
|-------------|-------------|-------------|-------------|------------------------|
|             |             |             |             | <a href="#">Verify</a> |
|             |             |             |             | <a href="#">Verify</a> |
|             |             |             |             | <a href="#">Verify</a> |
|             |             |             |             | <a href="#">Verify</a> |
|             |             |             |             | <a href="#">Verify</a> |
|             |             |             |             | <a href="#">Verify</a> |
|             |             |             |             | <a href="#">Verify</a> |

## Verify selling vehicle

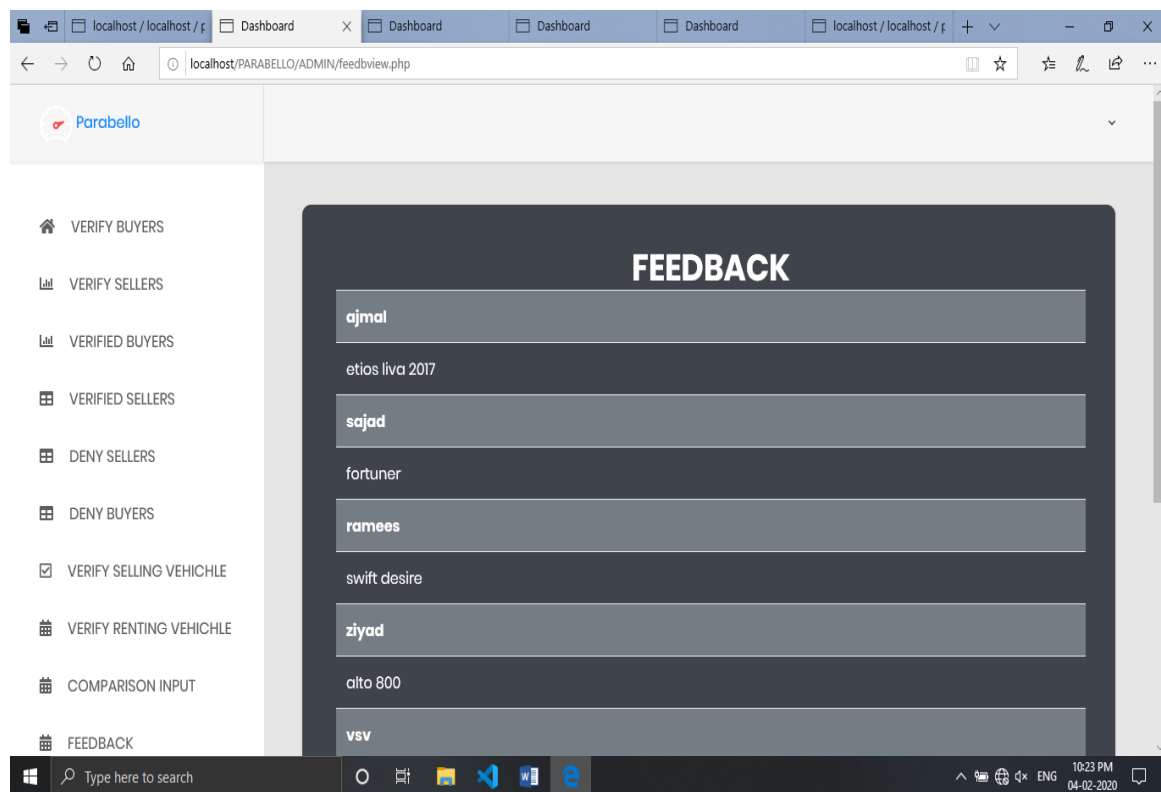


## Comparison input

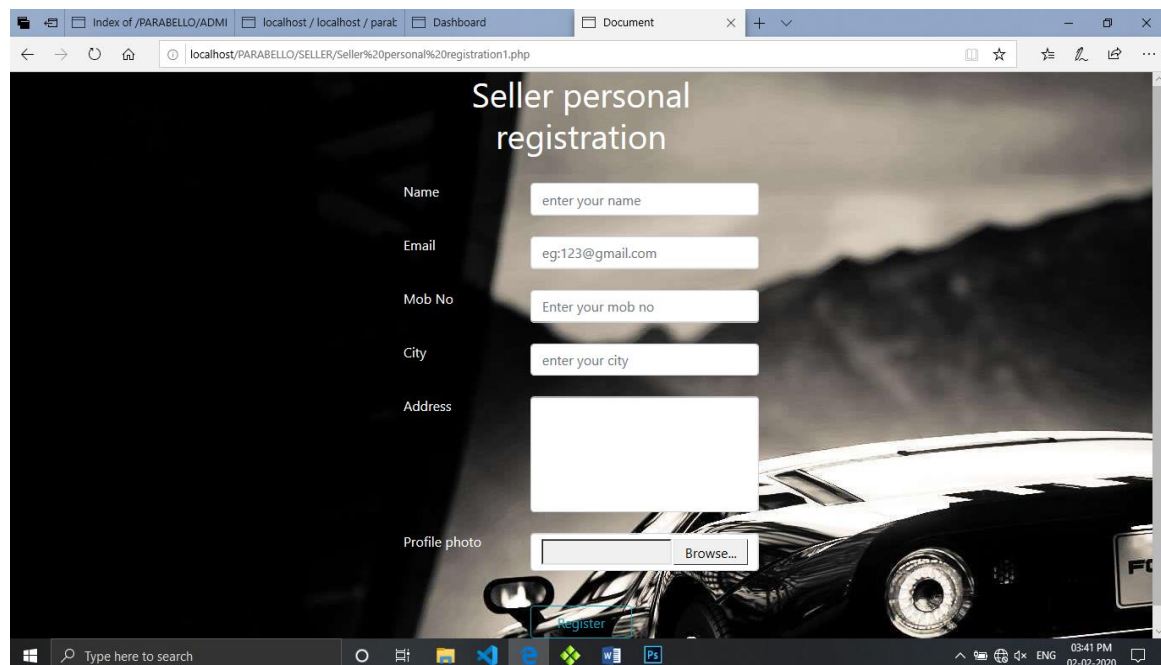




## Feedback



## Seller registration



## Seller login

The screenshot shows a web browser window with the address bar displaying `localhost/Parabello/SELLER/sellerlogin.php`. The page title is "Seller login". The login form consists of two input fields: "Seller id:" and "Password:", followed by a "login" button. The background of the page features a close-up image of a car's headlight and grille.

## Seller home



The screenshot displays the "Seller home" dashboard. On the left is a sidebar with the "Parabello" logo and a user profile "Mohammed Ramees5". The sidebar contains a "HOME" link and a list of actions: "Add Selling Vehicle", "Selling Vehicle", "New Rent", and "Renting Vehicle". The main content area is titled "ADD CAR DETAILS" and contains a form with the following fields: "Company", "Model", "Color", "KMs driven", "Fuel type", and "image". The "image" field includes a "Browse..." button. The dashboard is shown within a web browser window with the address bar displaying `localhost/Parabello/SELLER/cardetails.php`.

## Add selling vehicle

The screenshot shows a web browser window with the URL `localhost/PARABELLO/SELLER/cardetails.php`. The application header includes the Parabello logo and the user name 'Mohammed Ramees5'. A sidebar on the left contains navigation links: HOME, Add Selling Vehicle, Selling Vehicle, New Rent, and Renting Vehicle. The main content area is titled 'ADD CAR DETAILS' and contains a form with the following fields: Company, Model, Color, KMs driven, Fuel type, and Image. The Image field has two 'Browse...' buttons. The Windows taskbar at the bottom shows the time as 10:27 PM on 04-02-2020.

## Selling vehicle list

The screenshot shows a web browser window with the URL `localhost/PARABELLO/SELLER/selling%20vehicle%20list.php`. The application header includes the Parabello logo and the user name 'Mohammed Ramees5'. A sidebar on the left contains navigation links: HOME, Add Selling Vehicle, Selling Vehicle, New Rent, and Renting Vehicle. The main content area is titled 'Selling vehicle list' and displays a list of two vehicles. Each vehicle entry includes a car image, its details (Brand, Price, Model), and buttons for 'view', 'NONE', and 'Submit Query'.

| Image   | Brand             | Price         | Model                                | Actions   |
|---|-------------------|---------------|--------------------------------------|---|
|  | -BRAND:hyundai    | -PRICE:10000  | -MODEL:2013<br><a href="#">~more</a> | <button>view</button><br><button>NONE</button><br><button>Submit Query</button> |
|  | -BRAND:Etios Liva | -PRICE:110000 | -MODEL:2012<br><a href="#">~more</a> | <button>view</button><br><button>NONE</button><br><button>Submit Query</button> |

The Windows taskbar at the bottom shows the time as 10:28 PM on 04-02-2020.

## Add rent vehicle



The screenshot shows a web browser window with the URL `localhost/Parabello/SELLER/rentcardetails.php`. The page is titled "ADD RENT CAR DETAILS" and features a sidebar with navigation links: HOME, Add Selling Vehicle, Selling Vehicle, New Rent, and Renting Vehicle (which is selected). The main content area contains a form with the following fields:

- Company:
- Model:
- Color:
- KMs driven:
- Fuel type:
- image:  [Browse...](#)
- [Browse...](#)

The Windows taskbar at the bottom shows the time as 10:28 PM on 04-02-2020.

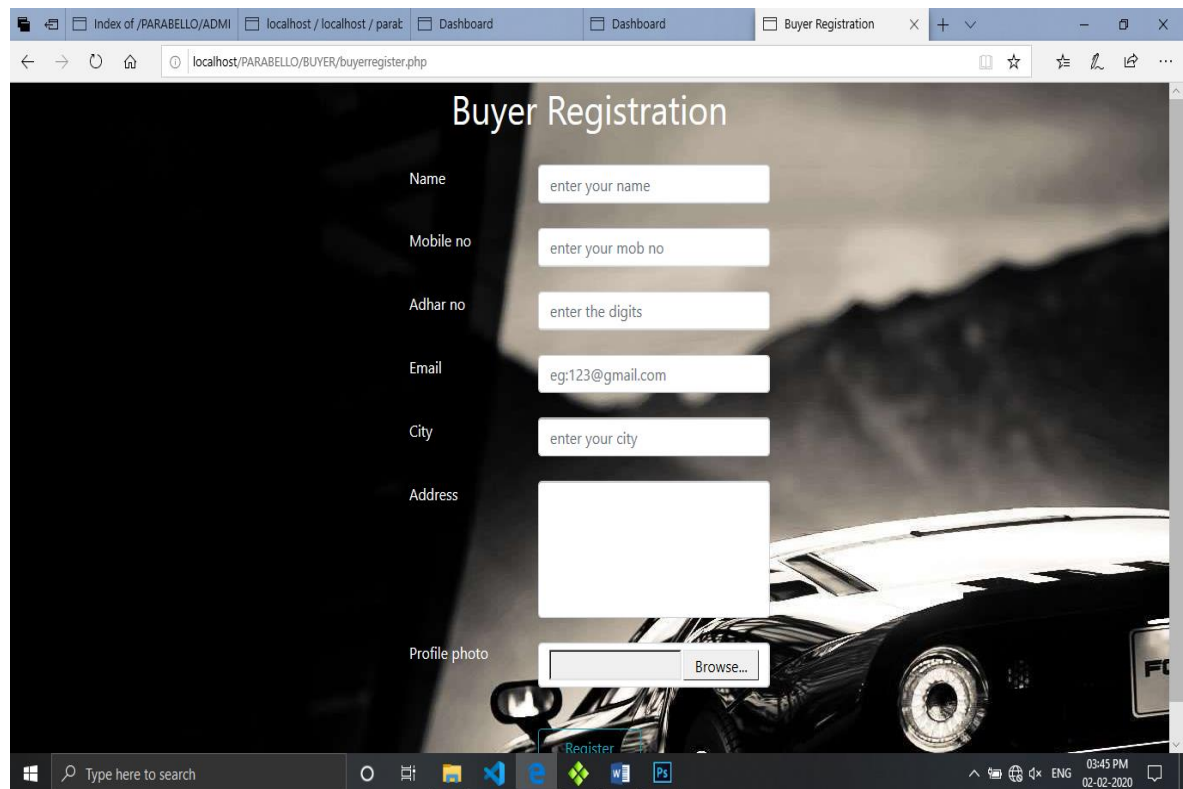
## Renting vehicle list

The screenshot shows a web browser window with the URL `localhost/Parabello/SELLER/renting%20vehicle%20list.php`. The page is titled "Seller detail list" and features a sidebar with navigation links: HOME, Add Selling Vehicle, Selling Vehicle, New Rent, and Renting Vehicle (which is selected). The main content area displays a list of vehicles for rent:

| Vehicle Image   | Details   | Action               |
|---|---|----------------------|
|  | <p>- BRAND:hyundai<br/>- PRICE:10000<br/>- MODEL:2013<br/><a href="#">-more</a></p> | <a href="#">view</a> |
|  | <p>- BRAND:hyundai<br/>- PRICE:10000<br/>- MODEL:2012<br/><a href="#">-more</a></p> | <a href="#">view</a> |

The Windows taskbar at the bottom shows the time as 10:29 PM on 04-02-2020.

## Buyer register



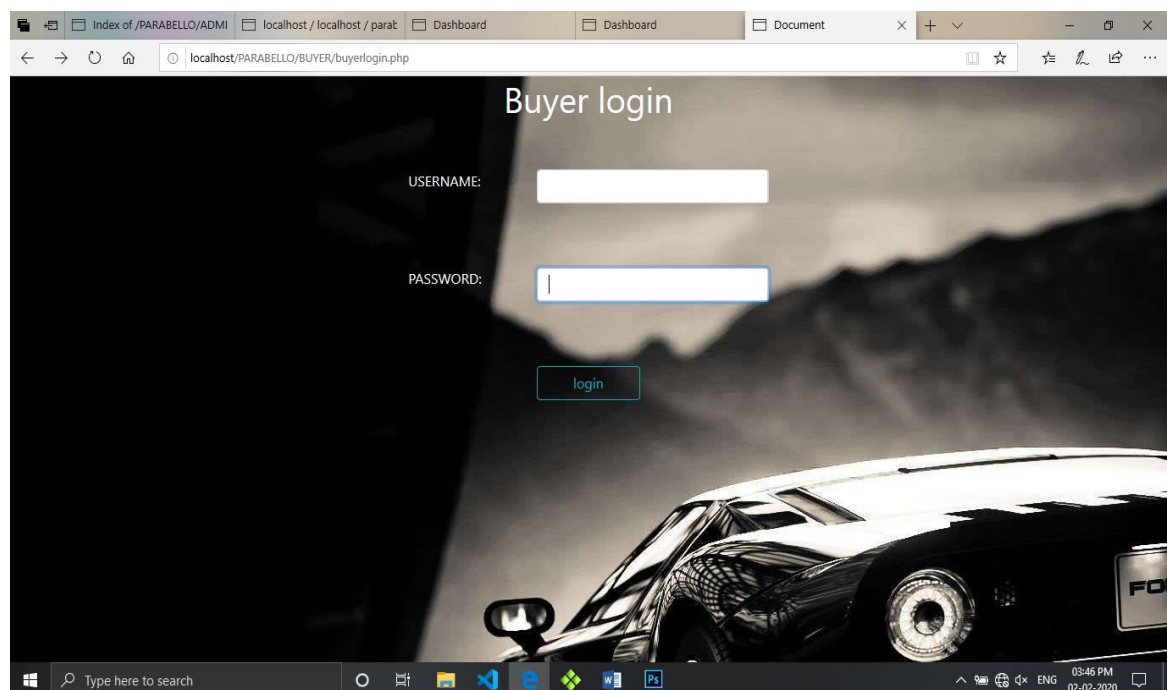
The screenshot shows a web browser window with the address bar displaying `localhost/PARABELLO/BUYER/buyerregister.php`. The page title is "Buyer Registration". The form contains the following fields and labels:

- Name: enter your name
- Mobile no: enter your mob no
- Adhar no: enter the digits
- Email: eg:123@gmail.com
- City: enter your city
- Address: (empty text area)
- Profile photo: (empty text area) with a "Browse..." button

The background of the form is a dark image of a car. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system clock in the bottom right corner displays "03:45 PM" and "02-02-2020".

## Buyer

## login

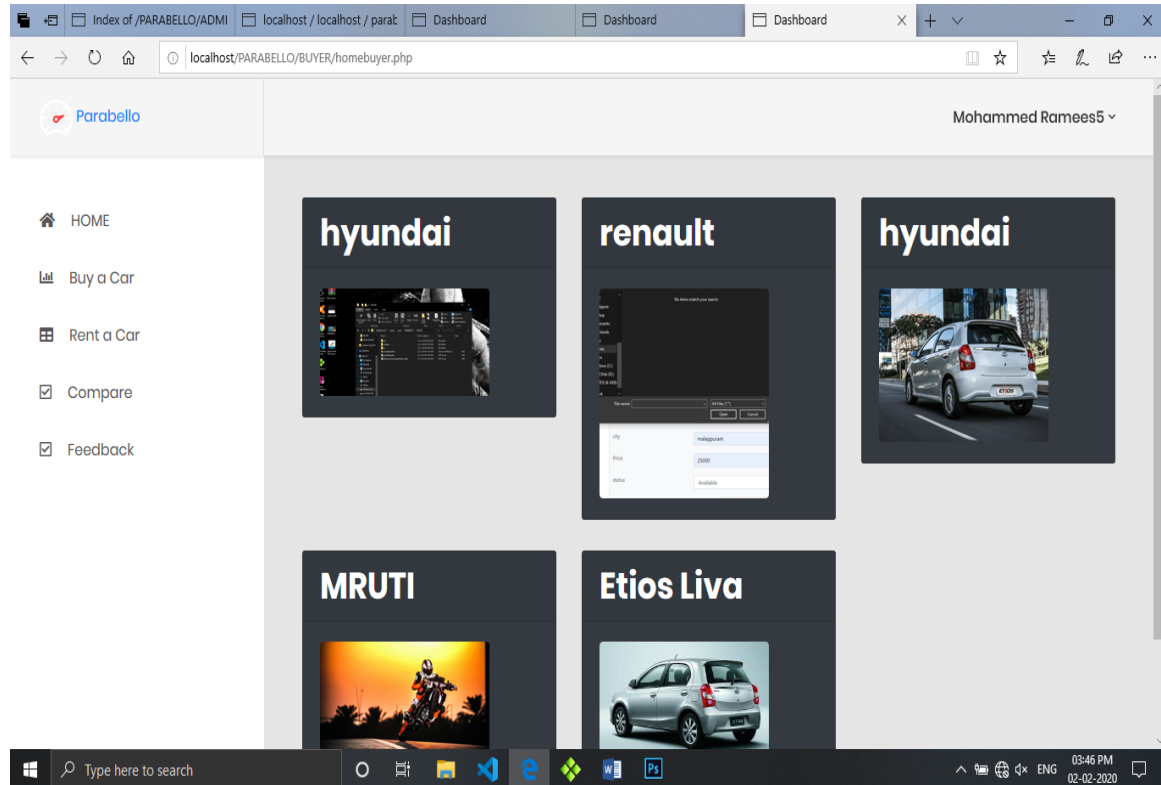


The screenshot shows a web browser window with the address bar displaying `localhost/PARABELLO/BUYER/buyerlogin.php`. The page title is "Buyer login". The form contains the following fields and labels:

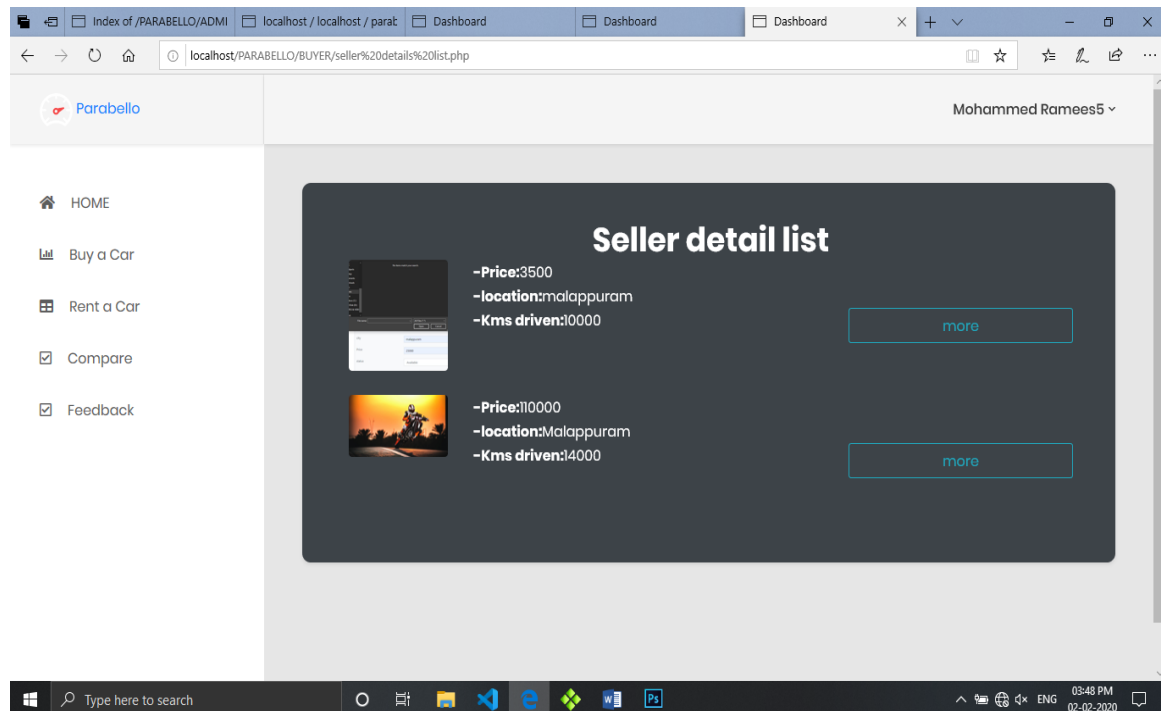
- USERNAME: (empty text input)
- PASSWORD: (empty text input)
- login: (blue button)

The background of the form is a dark image of a car. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system clock in the bottom right corner displays "03:46 PM" and "02-02-2020".

## Buyer home page



## Buying car list



## Rent car list

The screenshot shows a web browser window with the URL `localhost/Parabello/BUYER/renting%20vehicle%20list.php`. The page title is "Seller detail list". The user is logged in as "Mohammed Ramees5". The left sidebar contains the following menu items: HOME, Buy a Car, Rent a Car, Compare, and Feedback. The main content area displays a list of four cars for rent:

| Image | Price | Location   | Kms driven | Action               |
|-------|-------|------------|------------|----------------------|
|       | 3500  | malappuram | 15000      | <a href="#">more</a> |
|       | 10000 | malappuRAM | 15000      | <a href="#">more</a> |
|       | 10000 | malappuRAM | 15000      | <a href="#">more</a> |
|       | 10000 | malappuRAM | 125522     | <a href="#">more</a> |

## Compare

The screenshot shows a web browser window with the URL `localhost/Parabello/BUYER/BCompare%20page.php`. The page title is "Compare". The user is logged in as "Mohammed Ramees5". The left sidebar contains the following menu items: HOME, Buy a Car, Rent a Car, Compare, and Feedback. The main content area displays a form with two columns for selecting a vehicle and a model, and a "Submit Query" button:

| Choose available vehicle: | Select model: | Choose available vehicle: | Select model: |
|---------------------------|---------------|---------------------------|---------------|
| liva                      | 2012          | etios                     | 2012          |

[Submit Query](#)

## Feedback

The screenshot shows a web browser window with the URL `localhost/Parabello/BUYER/feedback.php`. The page features a sidebar with navigation links: HOME, Buy a Car, Rent a Car, Compare, and Feedback. The main content area is titled "Send your Feedback" and contains a form with a "NAME:" label and a text input field, followed by a "Feedback" label and a larger text area. The browser's address bar shows the local host path, and the Windows taskbar at the bottom indicates the time as 12:32 PM on 10-03-2020.

## Public user compare

The screenshot displays a web browser window with the URL `localhost/Parabello/PUBLIC%20USER/BCompare%20page.php`. The sidebar on the left includes links for Compare and Feedback. The main content area is titled "Compare" and features two columns of dropdown menus. The first column has "Choose available vehicle:" with a dropdown showing "liva" and "Select model:" with a dropdown showing "2012". The second column has "Choose available vehicle:" with a dropdown showing "etios" and "Select model:" with a dropdown showing "2012". A "Submit Query" button is positioned below the dropdowns. The browser's address bar shows the local host path, and the Windows taskbar at the bottom indicates the time as 10:32 PM on 04-02-2020.



## Public user feedback

The screenshot shows a web browser window with the Parabello logo in the top left corner. The browser's address bar displays the URL `localhost/Parabello/PUBLIC%20USER/feedback.php`. On the left side of the page, there is a sidebar with two checked items: ☒ Compare and ☒ Feedback. The main content area features a large, dark-bordered box with a yellow header that reads "Send Your Feedback". Inside this box, there is a form with two fields: a text input labeled "NAME:" and a larger text area labeled "Feedback". The Windows taskbar at the bottom shows the search bar with the text "Type here to search", several application icons, and the system clock indicating 12:35 PM on 10-03-2020.

Dashboard border.pdf localhost / localhost / parat localhost / localhost / phpM Dashboard

localhost/Parabello/PUBLIC%20USER/feedback.php

Parabello

☒ Compare

☒ Feedback

**Send Your Feedback**

NAME:

Feedback

Type here to search

12:35 PM 10-03-2020