**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Particulars** | **Page No.** |
|  | Synopsis | 1 |
| **1.** | **Introduction** |  |
|  | 1.1. Project Overview | 2 |
|  | 1.2. Description of modules | 2 |
| **2.** | **System Study** |  |
|  | 2.1. Existing system | 5 |
|  | 2.1.1. Drawbacks of existing system | 5 |
|  | 2.2. Proposed system | 5 |
|  | 2.2.1. Advantages of proposed system | 6 |
|  | 2.3. System Specification | 7 |
|  | 2.3.1. Hardware Specification | 7 |
|  | 2.3.2. Software Specification | 7 |
| **3.** | **System Design** |  |
|  | 3.1. Design Notations | 13 |
|  | 3.1.1. Data Flow Diagram | 13 |
|  | 3.1.2. ER Diagram | 15 |
|  | 3.2. System Design | 16 |
|  | 3.2.1. Database Design | 16 |
| **4.** | **System Testing and Development** |  |
|  | 4.1. Introduction | 21 |
|  | 4.2. System Testing | 21 |
|  | 4.3. System Implementation | 22 |
|  | 4.4. System Development | 23 |
| **5.** | **Conclusion and Future Enhancement** | 25 |
|  | **Appendix** | 26 |

**SYNOPSIS**

This project is entitled as **‘Courier Management System’**. The system is being used for day to day activities such as maintain employee details, booking a courier, maintain hub details, maintain company details, process data of employees and many other things. The main aim of this project is to computerize the maintenance of courier management. Through this Courier Service Management System, one can know about parcel details like picked up, unpicked, and delivered and so on.,

**MODULES:**

* Wish master Registration.
* Courier Allotment.
* Delivery Update.
* Pickup Allotment.
* Pickup Update.
* Work Analysis.

1. **INTRODUCTION**

**1.1. PROJECT OVERVIEW**

Modules of the Project are

* Wishmaster Registration.
* Courier Allotment.
* Delivery Update.
* Pickup Allotment.
* Pickup Update.
* Work Analysis.

**WISHMASTER REGISTRATION:**

The Admin logins into the application using his/her credentials and manually enters each and every wishmaster details like Name, Contact No., Bank Account details, PAN card details, Aadhar card details, father’s name, father’s contact no., Reference name, Reference details etc., And the User Id(which is primary key) is set to generate in automatic.

**PARCEL DETAILS:**

The Admin Logins to the Application using their credentials. And after successfully logged in., He/She enters the details of each and every Parcels. i.e, Customer Name, Order ID, Ordered Date, Maximum Retail Price of the Product, Address of the Customer, Address of the Sender, Contact details of the Customer. Here, Order Id is in Primary Key so that we can avoid the repetition of same Order IDs.

**COURIER ALLOTMENT:**

The Admin allots the courier to the user using their Individual User ID. The Admin enters the Registered Wishmaster details(User Id) and enters the search button., the Name of the Wishmaster is displayed. The Admin can check whether the User Id he entered is correct or not using their name. The Order Id of the product is displayed in the drop down list box, When the Admin selects any one from the list, the other respective details like Customer Name, Address, MRP is auto filled. Admin can check once and click on Allot button to allot the courier to the Wishmaster.

**DELIVERY UPDATE:**

The Wishmaster logins to the Application using their respective credentials. And after successfully logged in He/She views the allotted couriers to him. The Order IDs(with reference to the Primary key of Parcel Details table) of the allotted parcels are shown in drop down listbox. When the wishmaster selects one from the above, the other details are filled automatically and the Wishmaster only needs to enter the status, date and the received person’s name and clicks on the Update button to Update the Delivery.

**PICKUP ALLOTMENT:**

After the product is delivered, Some customers may initializes the return pickup. The Admin enters the Order Id of the product, the other details are filled automatically and the Admin enters the User Id of the Wishmaster(who is going to take the return pickup) and click on search to view their name. The Admin checks the name and clicks the Allot button to allot the pickup to the Wishmaster.

**PICKUP DELIVERY:**

The Wishmaster views what are the pickups they have assigned to him/her. The Order IDs of the products is shown in drop down listbox. The Wishmaster selects one from the list and the other details are filled automatically. Now, the Wishmaster just enters the condition and clicks on Update button to Update the Pickups assigned.

**WORK ANALYSIS:**

The Admin checks the performance of each and every person and enter the details like Total No. of OFD(Out For Delivery), Delivered parcels, Undelivered parcels, Total No. of Pickups assigned, Picked items, Unpicked items and Date. The Wishmaster can view the progression of their work daily, weekly or in between a particular dates by selecting the Dates and click on.

1. **SYSTEM ANALYSIS**

**2.1. EXISTING SYSTEM:**

In existing system, only the administrator can be able to access the day to day activities like wishmaster’s details, delivery rate of individuals, non-delivery rate of individuals, return pickup and so on. There is no other user except administrator.

**DRAWBACKS OF EXISTING SYSTEM:**

* Individuals cannot access their daily activities.
* Wishmaster’s cannot calculate their salaries.
* Wishmaster’s don’t know the production rate of their hub.
* Retrieval of the data is difficult.

**2.2. PROPOSED SYSTEM:**

The system to be developed for the company is fully automated with system developed using VB.Net as Front End and SQL Server as Back End. The Individuals can easily access their day to day production rate without depending on the hub administrator. And also the authorised user can be able to know the production rate of their Hub.

**ADVANTAGES OF THE PROPOSED SYSTEM:**

* Time saving consumer less time.
* Proposed system is user friendly interface.
* Tracking is made easier and fastly.
* Individual access for daily activities is made Easy.
* Retrieval of a particular data is Easy.

**2.3. SYSTEM SPECIFICATION**

**2.3.1. HARDWARE SPECIFICATION:**

|  |  |
| --- | --- |
| Processor | : Intel Core i3 (or) AMD RYZEN 3 |
| RAM | : 2 GB |
| Hard Disk | : 100 GB |
| Clock Speed | : 2.1 GHz |

**2.3.2. SOFTWARE SPECIFICATION:**

|  |  |
| --- | --- |
| Operating System | : Windows |
| Technology | : Microsoft Visual Studio |
| Front End | : VB.Net |
| Back End | : SQL Server |

**ABOUT THE SOFTWARE:**

The system is developed using Visual Basic.NET, which is a very popular Microsoft Product developed by Microsoft Corporation. This is one of the improved language from basic language. Visual Basic .NET includes a variety of open active controls for user interfaces to design application form.

Visual Basic .NET, the next generation of the Visual Basic language, is a fast and easy way to create .NET-based applications, including XML Web services and Web applications.

Visual Basic .NET has many new and improved features that make it a powerful object-oriented programming language, and overloading. Other new language features include free threading and structured exception handling. The language is designed with Rapid Application Development in mind, providing several tools to shorten development time.

This book introduces Visual Basic .NET language fundamentals and covers a variety of the base class libraries (BCL) provided by the .NET Framework.

**FEATURES OF THE COMMON LANGUAGE RUNTIME:**

The Common language runtime manages memory, thread execution, code execution, code safety verification, compilation and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

The runtime also enforces code robustness by implementing a strict type-and-code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

The runtime also accelerates developer productivity. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

**.NET Framework Class Library**

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

**Visual Basic.Net:**

Visual Basic .NET is an object-oriented computer programming language implemented on the .NET Framework. Although it is an evolution of classic Visual Basic language, it is not backwards-compatible with VB6, and any code written in the old version does not compile under VB.NET. The Visual Basic IDE is made up of a number of components given below

* Forms
* Toolbox
* Tabs
* Properties windows
* Solution explorer

**SPECIAL FEATURES IN VB.NET:**

* Modern, general purpose
* Object oriented
* Component oriented and Easy to learn
* Structured language
* It produces efficient programs
* It can be compiled on a variety of computer platforms
* Part of .Net Framework

**ABOUT SQL SEVER:**

Microsoft SQL Server is a relational database management system, or RDBMS, that supports a wide variety of transaction processing, business intelligence and analytics applications in corporate IT environments. It&#39;s one of the three market-leading database technologies, along with Oracle Database and IBM&#39;s DB2.Like other RDBMS technologies,

SQL Server is primarily built around a row-based table structure that connects related data elements in different tables to one another, avoiding the need to redundantly store data in multiple places within a database. The relational model also provides referential integrity and other integrity constraints to maintain data accuracy; those checks are part of a broader adherence to the principles of atomicity, consistency, isolation and durability – collectively known as the ACID properties and designed to guarantee that database transactions are processed reliably. The core component of Microsoft SQL Server is the SQL Server Database.

**SPECIAL FEATURES IN MICROSOFT SQL SEVER:**

* Enterprise: Designed for large enterprises with complex data requirements, data warehousing and Web-enabled databases. Has all the features of SQL Server and its license pricing is the most expensive.
* Standard: Targeted toward small and medium organizations. Also supports e-commerce and data warehousing.
* Workgroup: For small organizations. No size or user limits and may be used as the backend database for small Web servers or branch offices.
* Express: Free for distribution. Has the fewest number of features and limits database size and users. May be used as a replacement for an Access database.

1. **SYSTEM DESIGN**

**3.1. DESIGN NOTATION**

**3.1.1. DATA FLOW DIAGRAM**

**LEVEL 0:**

**3.2.1. DATABASE DESIGN**

**TABLES**

**Table Name: Employee\_Details**

**Primary Key: user\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Column Name** | **Data Type** | **Constraints** |
| 1. | name | nvarchar(50) | null |
| 2. | contact | decimal(18,0) | null |
| 3. | email | nvarchar(50) | null |
| 4. | father\_name | nvarchar(50) | null |
| 5. | father\_contact | decimal(18,0) | null |
| 6. | ref\_name | nvarchar(50) | null |
| 7. | ref\_contact | decimal(18,0) | null |
| 8. | user\_id | int | Primary Key |
| 9. | location | nvarchar(50) | null |
| 10. | pan\_card | nvarchar(50) | null |
| 11. | adhar\_card | decimal(18,0) | null |
| 12. | acnt\_num | decimal(18,0) | null |
| 13. | ifsc\_code | nvarchar(50) | null |
| 14. | password | Nvarchar(50) | null |

**Table Name: Parcel\_Details**

**Primary Key: order\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Column Name** | **Data Type** | **Constraints** |
| 1. | order\_id | int | primary key |
| 2. | customer\_name | nvarchar(50) | null |
| 3. | ordered\_date | date | null |
| 4. | cpd\_date | date | null |
| 5. | sender\_address | nvarchar(MAX) | null |
| 6. | receiver\_address | nvarchar(MAX) | null |
| 7. | MRP | int | null |
| 8. | mobile\_num | decimal(18,0) | null |
| 9. | alt\_mob\_num | decimal(18,0) | null |
| 10. | product\_details | nvarchar(MAX) | null |

**Table Name: Courier\_Allotment**

**Foreign Key: order\_id(in reference with “parcel\_details” table )**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Column Name** | **Data Type** | **Constraints** |
| 1. | user\_id | int | null |
| 2. | order\_id | int | Foreign Key |
| 3. | name | nvarchar(50) | null |
| 4. | mrp | int | null |
| 5. | allotment\_date | date | null |
| 6. | address | nvarchar(MAX) | null |

**Table\_Name: Delivery\_Update**

**Foreign Key: order\_id(in reference with “parcel\_details” table )**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Column Name** | **Data Type** | **Constraints** |
| 1. | user\_id | int | null |
| 2. | order\_id | int | Foreign Key |
| 3. | mrp | int | null |
| 4. | cus\_name | nvarchar(50) | null |
| 5. | delivered\_date | date | null |
| 6. | received\_by | nvarchar(50) | null |
| 7. | delivered\_at | nvarchar(50) | null |
| 8. | status | nvarchar(50) | null |

**Table Name: Pickup\_Details**

**Foreign Key: order\_id(in reference with “parcel details” table)**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Column Name** | **Data Type** | **Constraints** |
| 1. | order\_id | int | Foreign Key |
| 2. | cus\_name | nvarchar(50) | null |
| 3. | address | nvarchar(MAX) | null |
| 4. | init\_date | date | null |
| 5. | product\_details | nvarchar(MAX) | null |
| 6. | quantity | int | null |
| 7. | code | int | null |
| 8. | contact\_num | decimal(18,0) | null |

**Table Name: Pickup\_Allotment**

**Foreign Key: user\_id (in reference with “Employee\_Details” table)**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Column Name** | **Data Type** | **Constraints** |
| 1. | user\_id | int | Foreign\_key |
| 2. | name | nvarchar(50) | null |
| 3. | order\_id | int | null |
| 4. | cus\_name | nvarchar(50) | null |
| 5. | location | nvarchar(50) | null |
| 6. | reason | nvarchar(MAX) | null |
| 7. | date | date | null |

**Table Name: Pickup\_Update**

**Foreign Key: order\_id (in reference with “parcel\_details” table)**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Column Name** | **Data Type** | **Constraints** |
| 1. | user\_id | int | null |
| 2. | order\_id | int | Foreign key |
| 3. | cus\_name | nvarchar(50) | null |
| 4. | contact\_num | decimal(18,0) | null |
| 5. | address | nvarchar(MAX) | null |
| 6. | product\_details | nvarchar(MAX) | null |
| 7. | quantity | int | null |
| 8. | condition | nvarchar(50) | null |
| 9. | code | int | null |
| 10. | date | date | null |

**Table Name: Work\_Analysis**

**Foreign Key: user\_id(in reference with “Employee Details” table)**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Column Name** | **Data Type** | **Constraints** |
| 1. | user\_id | int | Foreign Key |
| 2. | name | nvarchar(50) | null |
| 3. | ofd\_parcels | int | null |
| 4. | delivered\_parcels | int | null |
| 5. | pickups\_assigned | int | null |
| 6. | picked\_items | int | null |
| 7. | UD\_parcels | int | null |
| 8. | unpicked\_items | int | null |
| 9. | date | date | null |

**4. SYSTEM TESTING AND DEVELOPMENT**

**4.1 INTRODUCTION**

The method of designing the program to produce the desired result is accomplished to the section. Programs are written and tested by using sample data: if the output produces good result was that the actual required output. All the facilities required for developing the new system acquired and program preparation is started. Quality assurance is an important step in software engineering. This overlaps with all the phases of development right from the requirement analysis.

This quality requirement phase of the software system must be clearly extracted during the requirement analysis and all the subsequent should be made biased to that, so that the final testing will become trivial and less expensive.To complete the cycle, there should be a post audit of the system to evaluate how well it performs and how well it meets the cost and performance specifications. The stages of definition, development and installation and operation can therefore be divided into smaller steps or phrases**.**

**4.2 SYSTEM TESTING**

**Unit Testing**

This testing method considers a module as single unit and checks the unit at interfaces and communities with other modules rather than getting into details at statement level. Here the module will be treated as BLACKBOX, which will take some inputs and generate output. Outputs for a given set of input combination are pre calculated and are generated by the module.

**Integration Testing**

Here all the pre-tested individual modules will be assembled to create a larger system and tests are carried out at system level to make sure that all modules are working with each other. This testing methodology helps in making sure that all modules which are running perfectly when checked individually and are also running cohesion with other modules. For this testing we create test-cases to check all modules once and then a generated test combination of test paths with the system to make sure that no path is making its way into chaos.

**Validation Testing**

Testing is major quality control measure employed during software development. Its basic function is to detect errors. Sub functions when combined may not produce than it is desired. Global data structures can represent the problems. To uncover errors that are associated with interfacing the objective is to make test modules and built a program structure that has detected by design. In a non-incremental integration all the modules are combined in advance and the program is tested as a whole. Here error will appear in an end-less loop function.

**4.3 SYSTEM IMPLEMENTATION**

The implementation phase focuses on change that is associated with error correction, adaptations required as the software's environment evolves, and changes due to enhancements brought about by changing customer requirements. Over time, the original environment (Ex., CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate change to its external environment.

Computer software deteriorates due to change, and because of this, preventive maintenance, often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. In essence, preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted, and enhanced. Software configuration management (SCM) is an umbrella activity that is applied throughout the software process.

**4.4 SYSTEM DEVELOPMENT**

Although there are a growing number of applications (such as decision support systems) that should be developed using an experimental process strategy such as prototyping, a significant amount of new development work continue to involve major operational applications of broad scope. The application systems are large highly structured. User task comprehension and developer task proficiency is usually high. These factors suggest a linear or iterative assurance strategy.

The most common method for this stage class of problems is a system development life cycle modal in which each stage of development is well defined and has straightforward requirements for deliverables, feedback and sign off. The system development life cycle is described in detail since it continues to be an appropriate methodology for a significant part of new development work.

**COURIER MANAGEMENT SYSTEM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case No.** | **Test Case** | **Check Item** | **Steps to Execute** | **Test Data/ Input** | **Expected Results** | **Actual Results** |
| TC-001 | CMS Admin Login Page | Form data about Admin Id and Password Details. | Login Button | Admin ID and Password | Error message displayed for invalid Login. | Same as expected Result. |
| TC-002 | CMS Admin Login Page | Form data about Admin Id and Password Details. | Login Button | Admin ID and Password | Admin menu displayed for valid Login. | Same as expected Result. |
| TC-003 | CMS work analysis | No. of OFD must be equal to the Delivered and undelivered parcels | Submit Button | No. of OFD, Delivered and Undelivered parcels. | Successfully saved message should pop up. | Same as expected result. |
| TC-004 | CMS work analysis | No. of OFD must be equal to the Delivered and undelivered parcels | Submit Button | No. of OFD, Delivered and Undelivered parcels. | Error message should pop up. | Same as expected result. |

**5. CONCLUSION AND FUTURE ENHANCEMENT**

**CONCLUSION:**

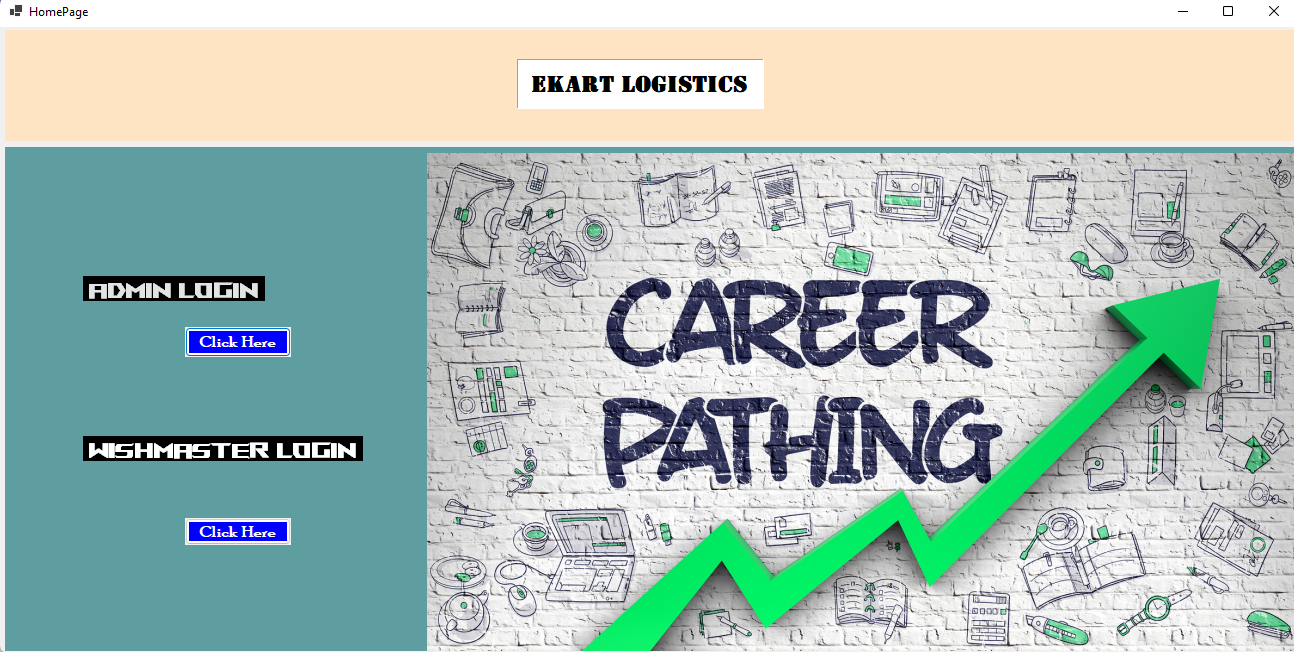
The COURIER MANAGEMENT SYSTEM app is useful for Courier allotting and delivery updating and also the WishMaster can able to view the each person’s individual work analysis. Administrator, WishMaster are the users of this project. Admin maintains the parcel details and courier allotment and the WishMaster maintains the delivery update.

**FUTURE ENHANCEMENT:**

The following features are to be included in the future:

* Payment method to be included in future using the UPI apps.
* Customer’s Location based GPS tracking facilities to be included.

**COURIER MANAGEMENT SYSTEM**

****

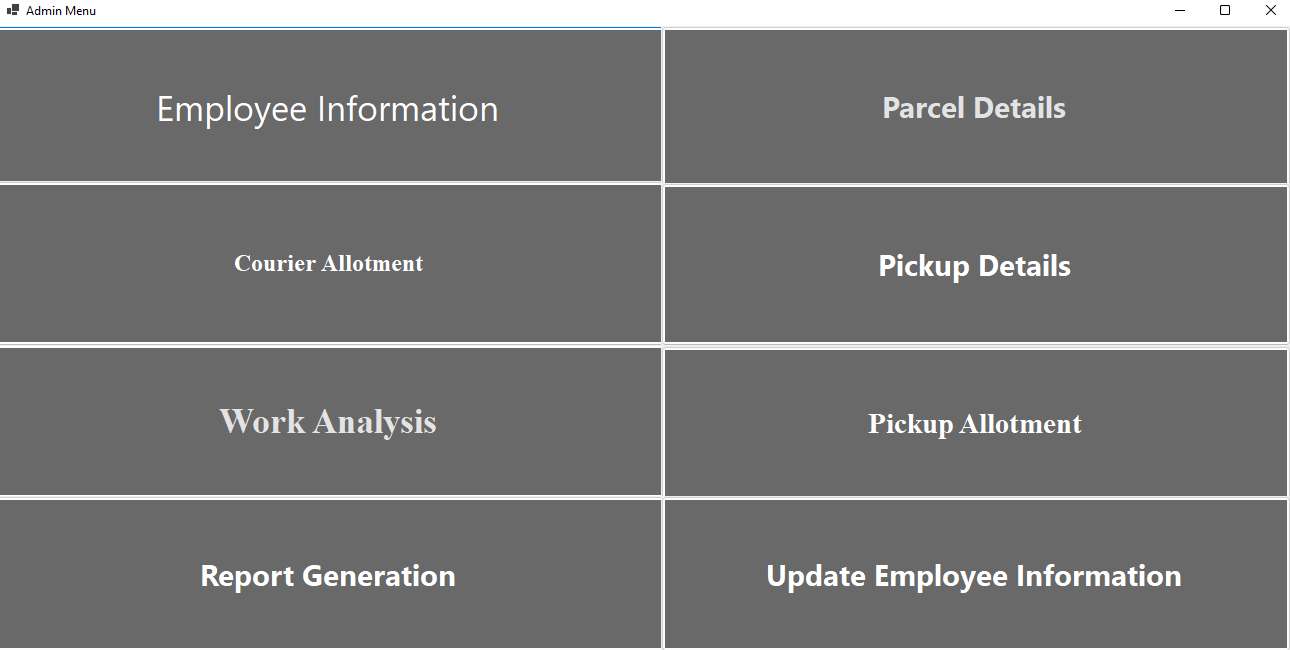
**HOME PAGE**

**COURIER MANAGEMENT SYSTEM**

****

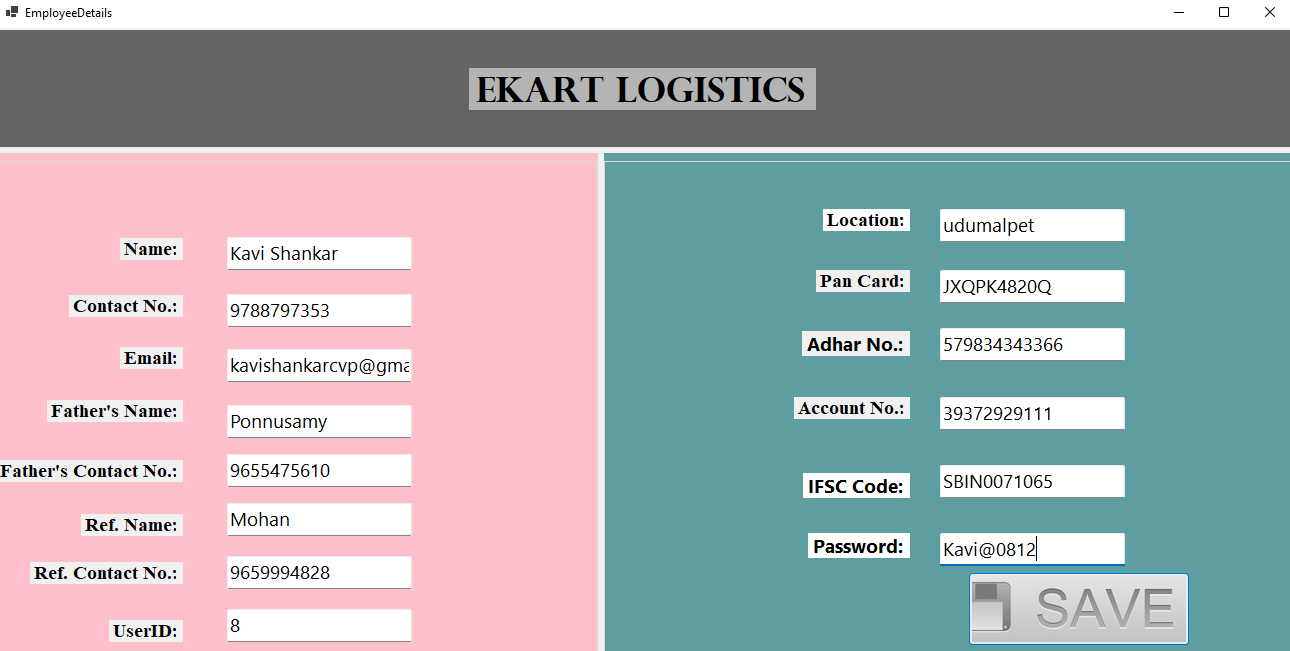
ADMIN LOGIN

**COURIER MANAGEMENT SYSTEM**

****

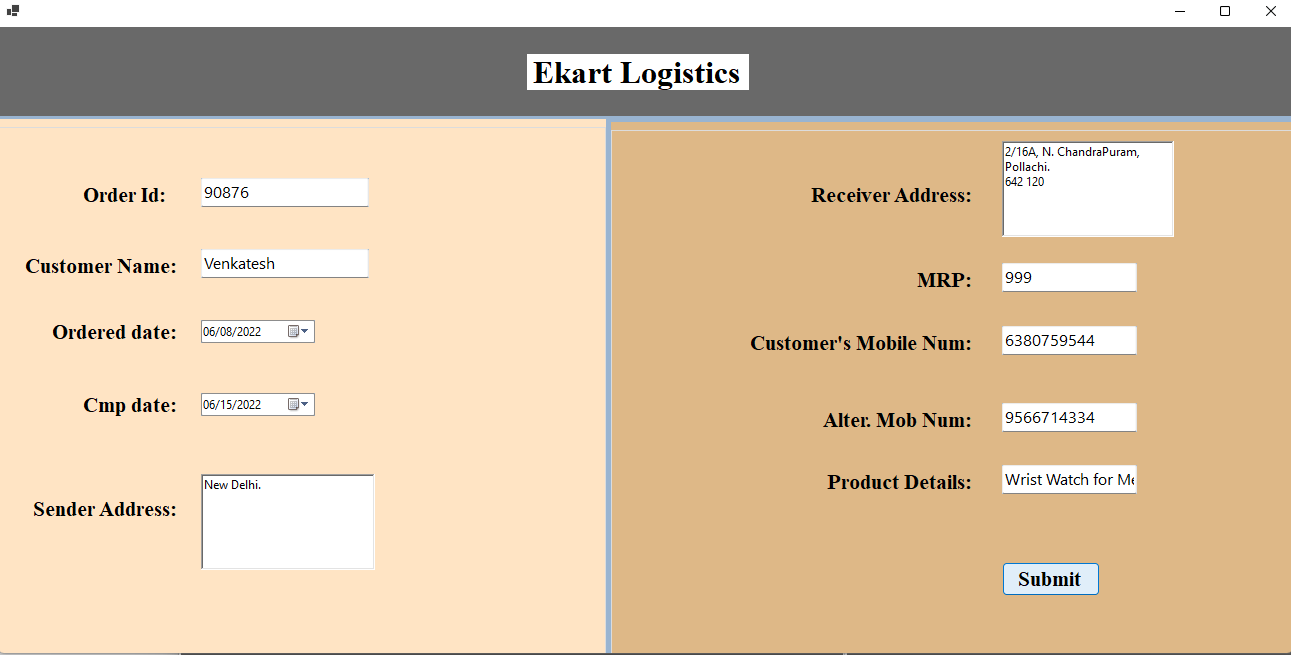
ADMIN MENU

**COURIER MANAGEMENT SYSTEM**



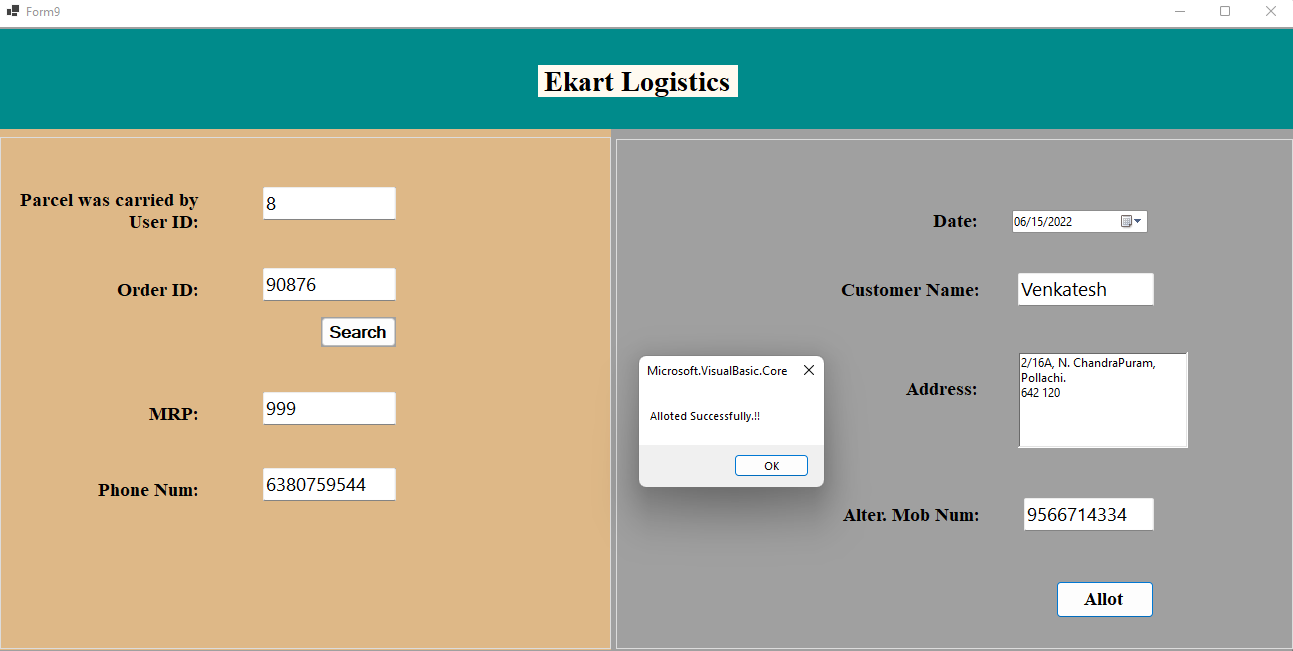
EMPLOYEE INFORMATION

**COURIER MANAGEMENT SYSTEM**



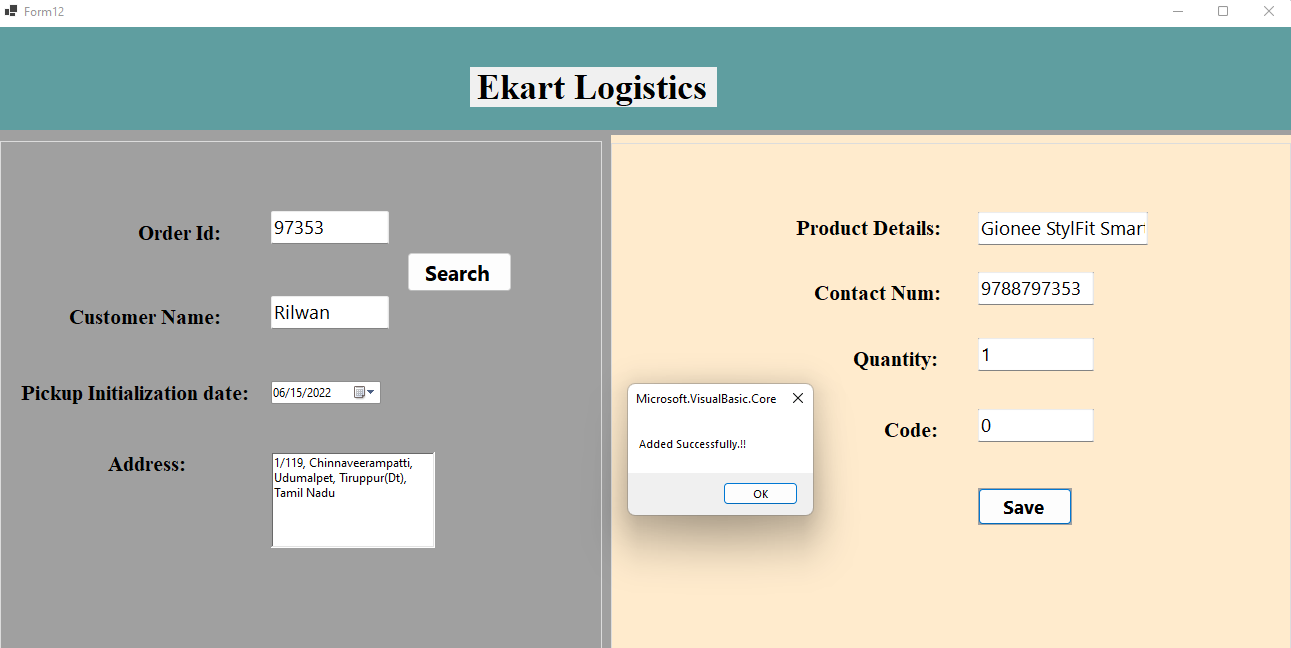
PARCEL DETAILS

**COURIER MANAGEMENT SYSTEM**



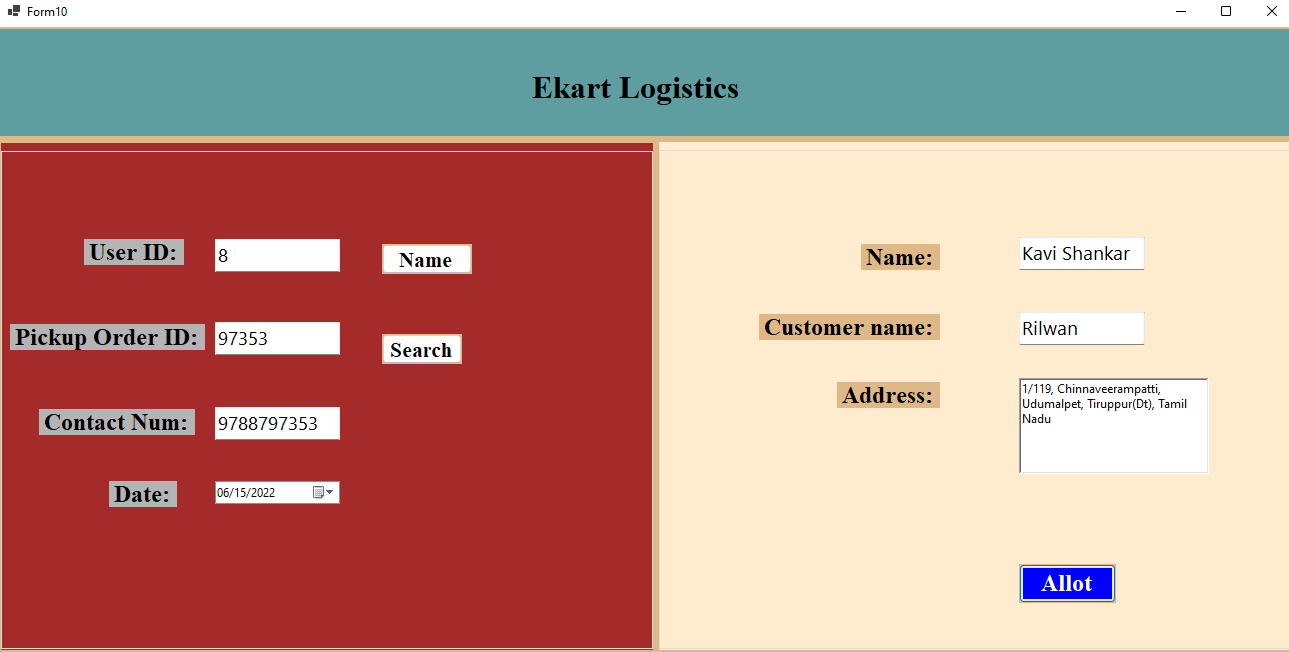
COURIER ALLOTMENT

**COURIER MANAGEMENT SYSTEM**



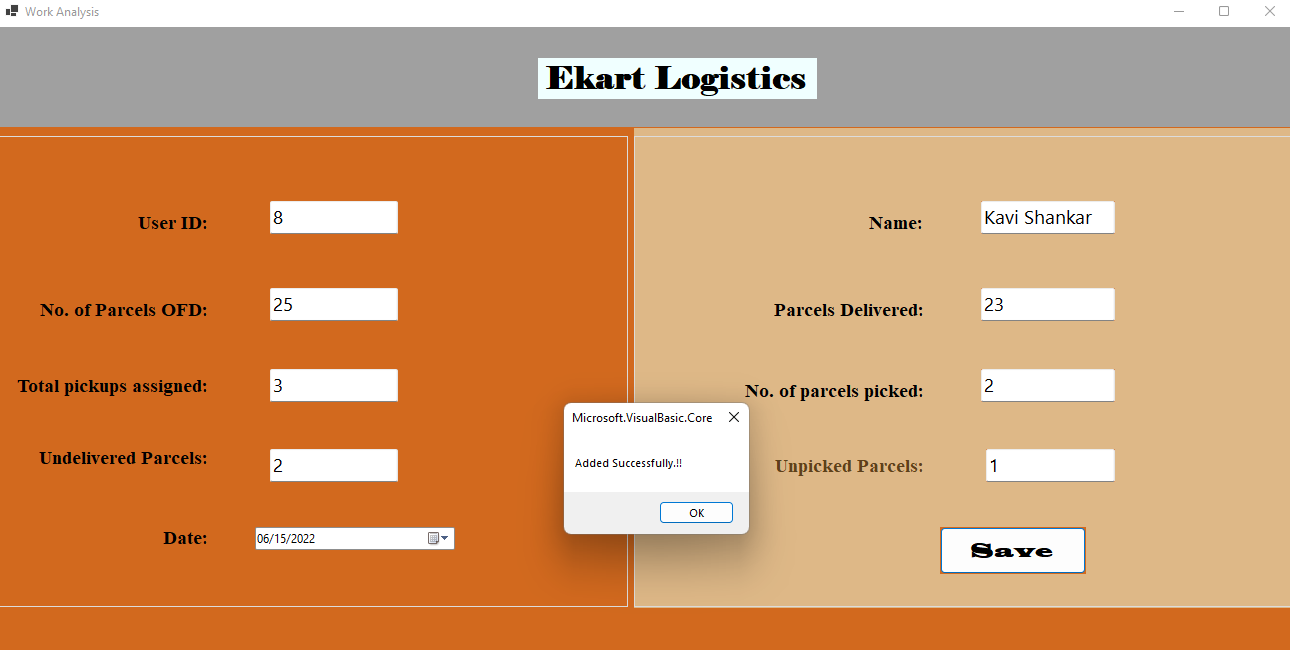
PICKUP DETAILS

**COURIER MANAGEMENT SYSTEM**



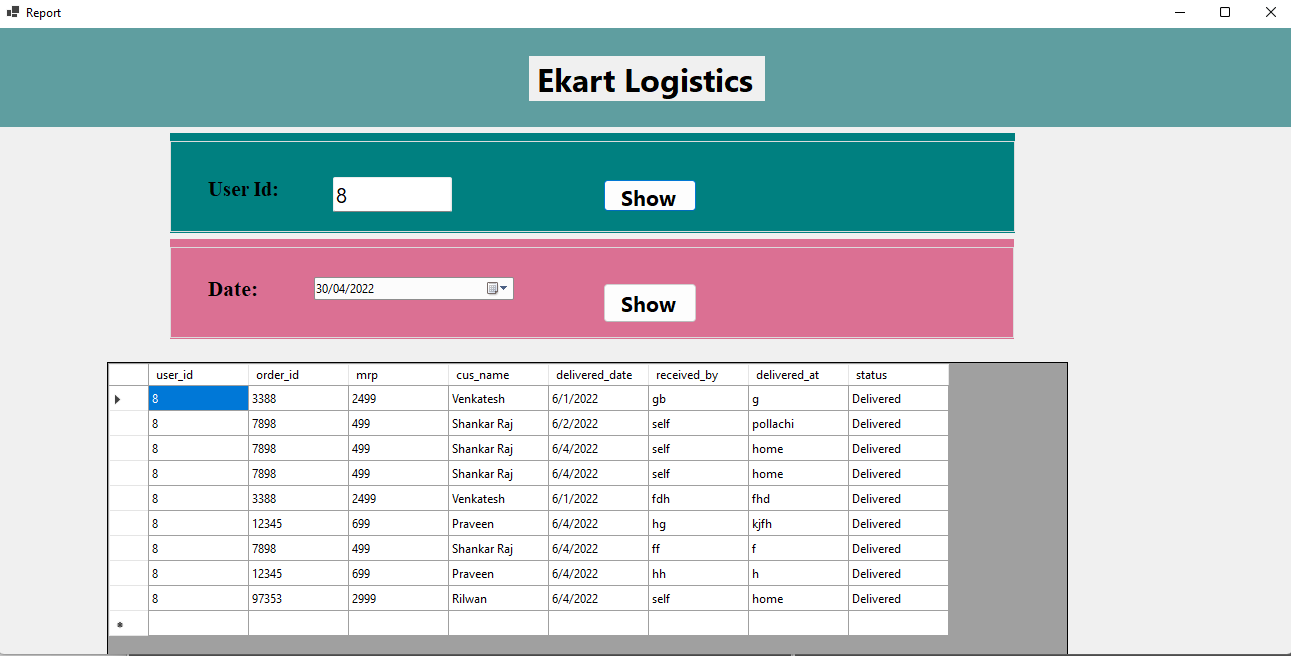
PICKUP ALLOTMENT

**COURIER MANAGEMENT SYSTEM**



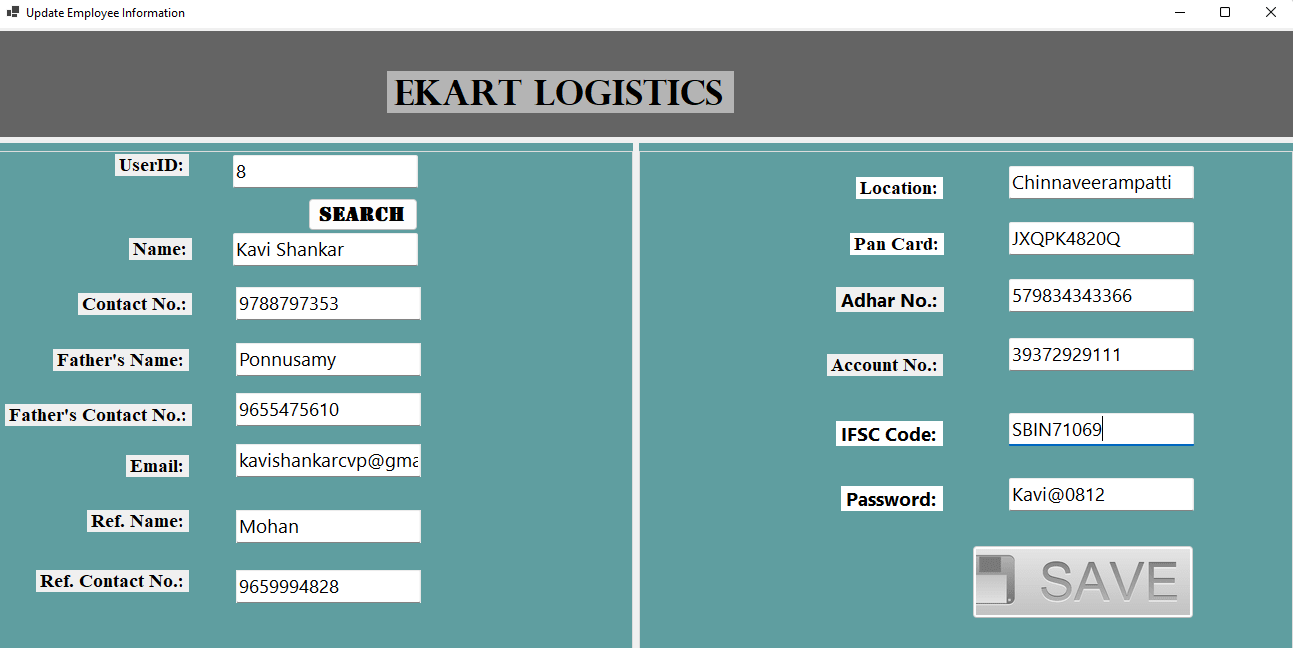
WORK ANALYSIS

**COURIER MANAGEMENT SYSTEM**



REPORT GENERATION

**COURIER MANAGEMENT SYSTEM**



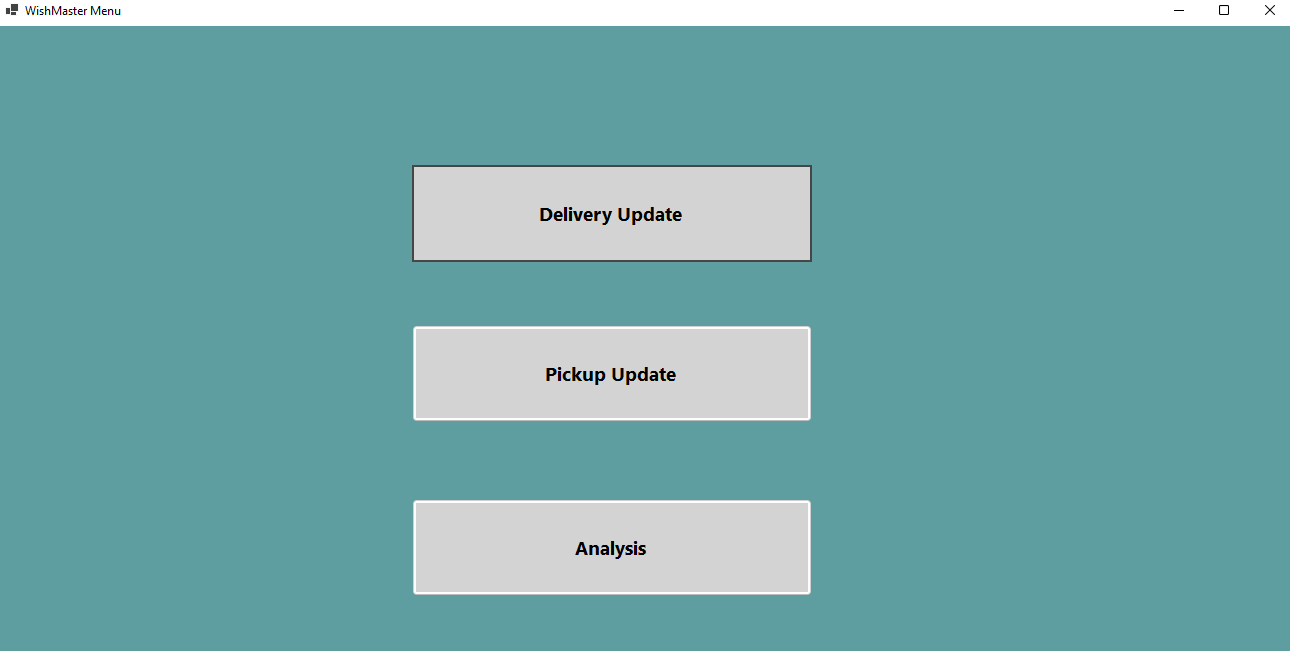
UPDATE EMPLOYEE INFROMATION

**COURIER MANAGEMENT SYSTEM**

****

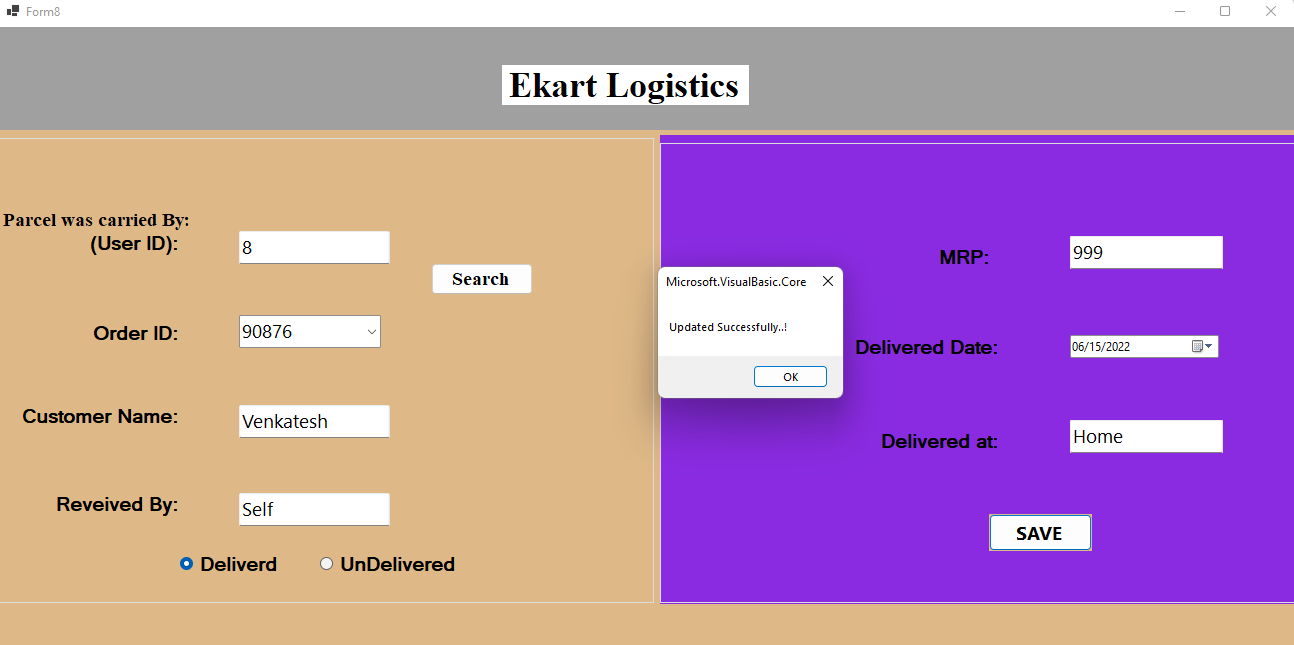
WISHMASTER LOGIN

**COURIER MANAGEMENT SYSTEM**

****

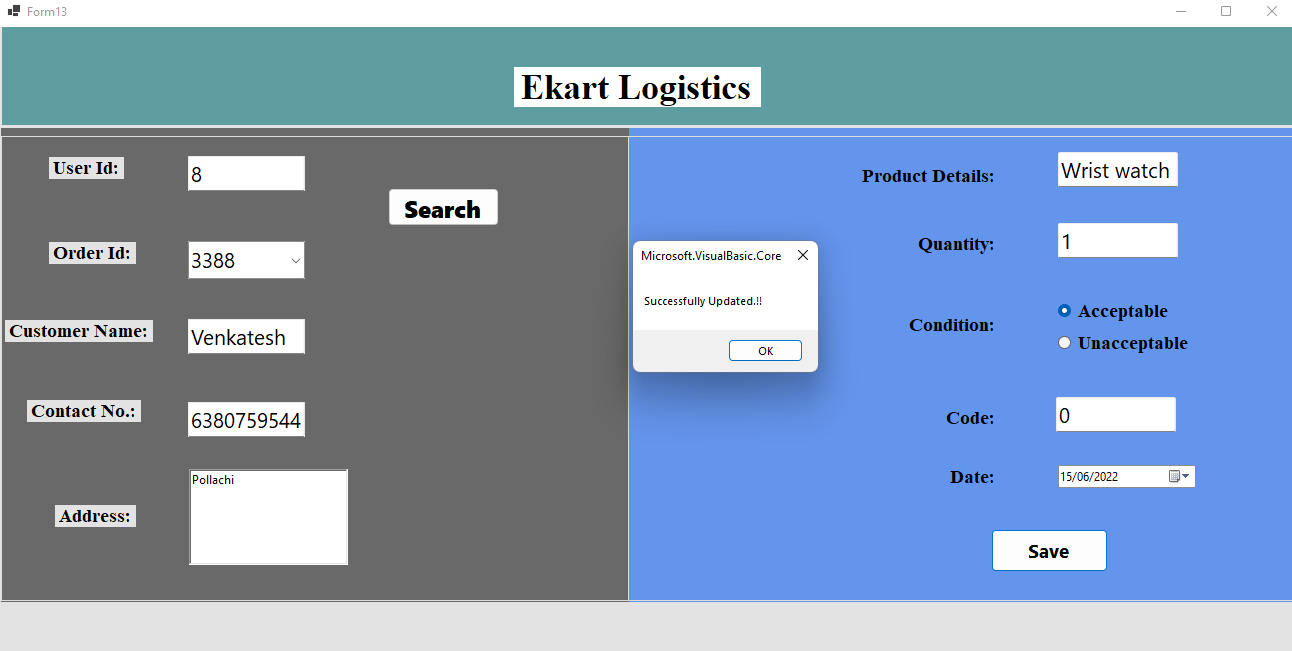
WISHMASTER MENU

**COURIER MANAGEMENT SYSTEM**



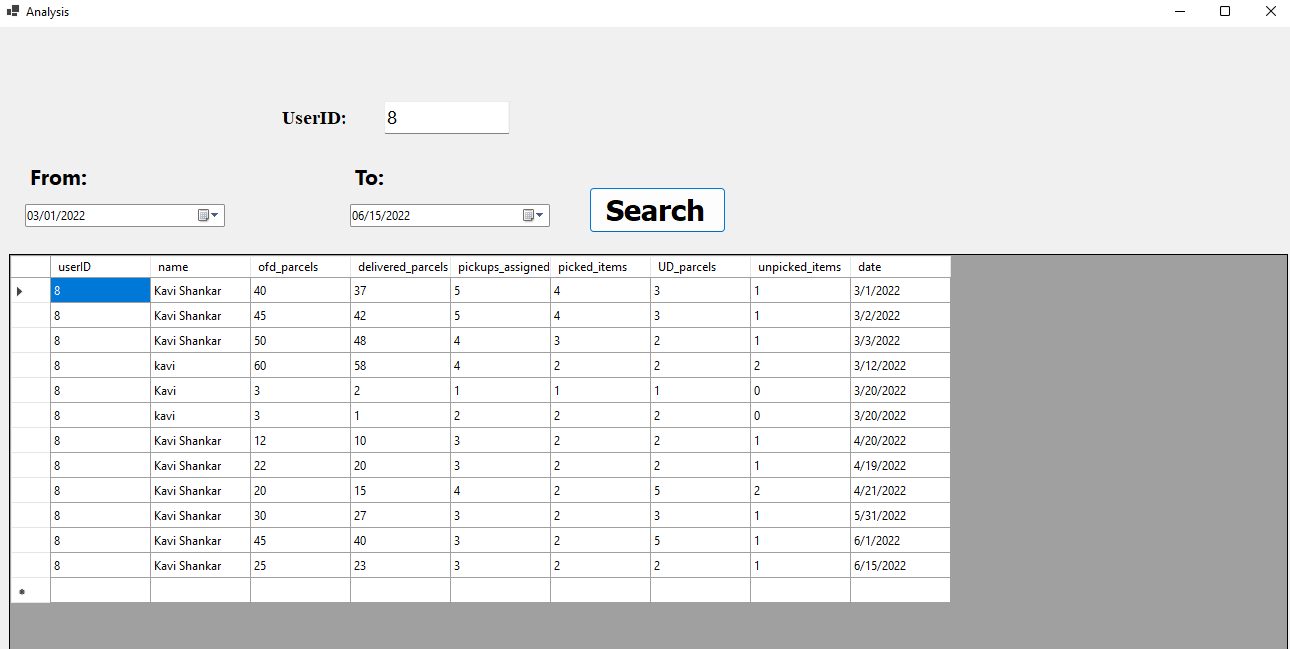
DELIVERY UPDATE

**COURIER MANAGEMENT SYSTEM**



PICKUP UPDATE

**COURIER MANAGEMENT SYSTEM**



ANALYSIS