#### A PROJECT REPORT

**ON**

**HOME LOAN WEBSITE**

**Submitted in the Partial fulfillment of requirement**

**For the award of degree**

**In**

**Bachelor of Computer Application**

**Of**

**BANGALORE UNIVERSITY**

**Submitted by**

**Names: KEERTHAN.D**

**Register No’s: 19SKSB7020**

**Under the Guidance of**

**Prof.Sreenivasa H V**

**Asst.Professor**

**School of Information Technology**

**Program-BCA**



**School of Information Technology**

**Bachelor of Computer Application**

### AIMS, BANGALORE

**Accredited by NAAC with‘A’Grade**

**Bangalore-560 058**

**CERTIFICATE of COMPLETION**

This is to certify that **KEERTHAN.D** bearing University Registration No’s:**19SKSB7037** has satisfactorily completed the Sixth Semester BCA Project-titled **“HOME LOAN WEBSITE“**. This report is submitted in partial fulfillment of the requirements for the award of the Degree in Bachelors of Computer Applications(BCA) as prescribed by Bangalore University, Academic Year 2021-22.

Project Guide Program Director

**Prof.Sreenivasa H V Dr.Govindaraj Pandith T.G**

Principal

**Dr.Kiran G Reddy**

**1st Stage,1st Cross,Peenya Bangalore-560 058,India**

**Tel:+91-80-65679113,28376430/28390434/41179588/41253496 Mobile:+91-09886021224 Fax:+91-8028378265**

1. **Mail:admissions@theaims.ac.in,principal@theaims.ac.in URL:**[**www.theaims.ac.in**](http://www.theaims.ac.in)

**UNDERTAKING BY THE STUDENT**

I have been working on the project entitled HOME LOAN WEBSITE from 25/05/2022 to 23/08/2022. I affirm that the originality and authenticity of the final year, Sixth semester project to be undertaken will be upheld.I declare that the report I submit represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the sources.Also,that I have adhered to all principle and protocols of AIMS, Bangalore and Bangalore University with academic honesty and integrity and have not misrepresented or fabricated any idea/data/fact in the submission.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Student

Date:

Student Name :KEERTHAN.D

Student ID :BCA19048

Registration No. :19SKSB7020

Contact No. :8197542097

Email Address :keerthanrao005@gmail.com

Course Name :Bachelor of Computer Application.

Year of Study :2019-2022

Area of Study :E-COMMERCE

Proposed Project Title :HOME LOAN WEBSITE.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Supervisor

Name:Prof. Sreenivasa H V

Date:

**ACKNOWLEDGEMENT**

With immense pleasure,I take this opportunity to express our gratitude to the beloved Principal and CEO **Dr.Kiran G Reddy** for providing valuable guidance and support during the course.

I would like to thank **Dr.Govindaraj Pandith T.G**, Director, School Information Technology- BCA Program for his unending support and encouragement during the development of the project.

I would like to acknowledge the interest and the support extended by our project guide **Prof.Sreenivasa H V**, Asst.Professor, School of Information Technology, BCA Program, to make this project implementation successful.

Finally, I wish to thank every individual who helped us directly or indirectly in making the project a grand success.

**KEERTHAN.D**

**19SKSB7020**

**INDEX**

**TABLE OF CONTENTS:**

**CHAPTER 1:Abstract……………………………………………..……….....01**

**1.1 Problem Definition.**

**1.2 Objectives.**

**1.3 Existing system.**

**1.4 Proposed system.**

**1.5 Hardware Requirements.**

**1.6 Software Requirements.**

**CHAPTER 2:Software Requirement and Specification…………………....…05**

**2.1 Introduction.**

**2.2 Purpose**

**2.3 Scope.**

**2.4 Definition and Abbreviation.**

**2.5 Reference.**

**2.6 Overview.**

**2.7 General Description.**

**2.8 User Characteristics.**

**2.9 General Constraints.**

**2.10 Functional Requirements.**

**2.11 Non-Functional Requirements.**

**2.12 Appendix.**

**CHAPTER 3:Detailed Design…………………………………...…..……..11**

**3.1 Data Flow Diagram(DFD).**

**3.2 Entity Relation.**

**3.3 Entity Relationship Diagram.**

**CHAPTER 4:Designing……….…………….…………………………..…..20**

**4.1 Database Design.**

**4.2 Basic Database Concepts for Opening section.**

**4.3 Tables.**

**CHAPTER 5:Coding………………………………………………………..26**

**5.1 PHP,CSS,HTML**

**5.2 Coding.**

**CHAPTER 6:Testing………………………………………………………..96**

**6.1 Introduction.**

**6.2 Software Testing types.**

**6.3 System Implementation.**

**CHAPTER 7:Form/Screen Layout/Output…………………..….………..99**

**7.1 Main Page**

**7.2 Admin Page.**

**7.3 Sign-up Page.**

**7.4 Registration Page.**

**7.5 Loan Application Page.**

**7.6 LoginPage.**

**7.7 Dashboard Page.**

**7.8 Request Page.**

**7.9 Help Page/FAQ.**

**CHAPTER 8:Conclusion…………………….……………..…………….109**

**CHAPTER 9:Future Enhancement.…………….….…………..……..…..111**

**CHAPTER 10:Appendix….…………………..…………….……………..113**

**CHAPTER 11:Bibliography...……………………..………………………115**

**ABSTRACT**

**CHAPTER-1**

**ABSTRACT**

* **Introduction:**
* HOME LOAN WEBSITE.
* This project of “**Home loan Website**” is developed in HTML/CSS with My Sql Database. Developed in Visual Studio Code. It is available for download on Windows, macOS and Linux based operating systems. This is a Home Loan Management Website that helps users to get loan to the citizens of India. It will help users to manage all banking and loan related activities in a single platform.
* We are using Php MySql realtime database so that we can update the content of the website in case of any changes made in the future.
  1. **Problem Definition:**

In case the customer could not repay the debt your lender might have the right to take something that you own, such as your car, if you have a secured loan. your lender can report a missed payment to the credit bureaus, which could mean it will show up on your credit history and could hurt your ability to get credit in the future

* 1. **Objective:**

The main purpose of a housing finance is to empower you to own your dream home. A home provides a long-term security and moreover a real estate property, usually, appreciates often than the other way around. A real estate is an investment that will give you good returns in the coming future.

**1.3 Existing System:**

There are quite several websites, blogs or apps regarding different types of loans in the world. It deals with various themes-based loans as well. But still there isn't any proper specified website for Home Loan with proper detailed description of loan ideas and how to take the steps and all. That is what we are going to provide through our website.The current available websites are not user friendly and are difficult to navigate through

**1.4 Proposed System:**

Our website "Home Loan Website" is based on only Home Loans which will contain all the specific details about each step to go through. Providing a loan should be a simple process. One should check the client’s eligibility to get the loan and then approve or deny the loan. Once approved, the customer should receive the funds.We will provide a website that is user friendly and can be navigated through with ease.

* The Proposed System has got the following advantages over the existing system.
* Manage the complete record of customer.
* Date wise reports on loan with customer details.
* Easily Maintainable and Update-able.
* Backup
* Multi-Level Reports.
* One Software deals with customer and loan details.
* Easy to handle.
* Privacy for the information and reliable.
* Help.

**1.5 Hardware Specification:**

* Client Side:RAM :512 MB.
* Hard disk: 10 GB.
* Processor :1.0 GHz.
* Server side:
* RAM : 1 GB.
* Hard disk : 20 GB.

**1.6 Software Specification:**

* Operating System : Windows 11.
* Front-End : PHP,HTML,CSS,JS.
* Connection-oriented: phpMyAdmin, XAMPP Server(Apache,MySQL)
* Database : SQL Server.

**Software Requirement and Specification.**

**CHAPTER-2**

**Software Requirement and Specification.**

**2.1 Introduction**

**2.2 Purpose**:

The Main Objective of this project is to develop online Web-based Application named “FAST TRACK RENTALS” which is to provide a detailed overview of our software product “Home Loan Website”,its parameters and goals. This document will give the user a brief description about the software.

**2.3 Scope:**

* “Home Loan Website” will provide the users with all the required details for a loan to the Customers. Loan management has the main scope of making things eaiser by depending on requirements, these programs can assist in part or whole. The software can help with customer lending loans by collecting the details and they do provide lenders with accurate statements and reports.
* The registration of loan is done by the customer itself along with respective details.
* Calculates the total amount and generates bill according to Days and KMS.
* We could view the customer history and overall bookings of the cars.
* The system also calculates the additional extra charges for crossing due dates.
* Provides the facility for Customers and Admin Login/Register.

**2.4 Definitions and Abbreviations:**

SQL:Structured Query Language.

VS:Visual Studio

JS: JavaScript.

HTML: Hypertext Markup Language.

CSS: Cascading Style Sheet.

PHP: Hypertext Pre-processor.

DFD:Data Flow Diagram.

ER:Entity Relationship.

IDE:Integrated Development Environment.

SRS:System Requirements Specifications.

**2.5 Reference:**

* GeeksforGeeks.
* Git Hub.
* Wikipedia.
* Stack Overflow.

**2.6. Overview:** Home Loan Website Using PHP and MYSQL is a Web based Application.The to give main objective is to provide good interaction & communication facilities between customers and the Adminstrator.Home Loan website has been designed to online the back office activites of bank and finance company that offers home loan.The Administrator manages the customer information database more efficiently,Loan details,inquiry and intrest rate information.Admin can use a calculator exact payable loan amount for the customer and generate all these work info as a report of each customer

**2.7 General Description:**

***Product Perspective:*** This is a Online based Web-Application which provides the customer a User-Friendly Interface in giving loans to the customer, Also it displays the available dteails for loan. Customers can fill up all the necessary fields about the information and get loans. The Home loan Website deals with the housing details, query about the loan details as well as EMI details in their profile.

1. ***Product Functions:***

* ThisInterest rates and the loan details are also available at the click of a mouse.
* Customer can apply for a loan and after approved it they can track their details from online.
* Using with this system admin can find customer easily and it’s a paperless system so workload is reduced.
* The decision process becomes faster and more consistent
* Provides good communication for the customer

In this system there are used EMI(Equated Monthly Installment) calculators

* **2.8 User Characteristics:**
* This Website provides detail about the customers, their loan details, EMI details and its rate details.
* Customer can apply for a loan and after approved it they can track their details from online.
* The decision process becomes faster and more consistent
* Customer can use the system easily and also customer can view any query about loan details as well as EMI details

**2.9 General Constraints:**

* This application is completely depended on web browser.
* This application have a restricted functionality.
* Web application generally lacks the quality control feature.

1. **Assumption and dependencies:**

All the data will be accurate and up-to date as the Integrated Development Environment used for coding is Visual Studio Code with HTML,CSS,JS,PHP as front end and MySQL as back end with XAMPP and PhpMyAdmin as connection-oriented between front end and back end.

**2.10 Functional Requirement:**

* **Client**: Allows to enter the client details
* **Admin**: Maintaining the database and also **Login and Register** : Allows User/Employee to create a User name and password for their account.
* availability of the loans

**2.11**  **Non-Functional Requirement:**

* + - **Usability requirements**: The system allows intended users to access the software from the pc using web application. The system uses browser as interface. No special training is required.
    - **Availability requirements**: The system can be accessed by the intended users at any given time. The system will always be operational, round the clock.
    - **Accurac**y: The system is designed in order to provide accurate information taking into consideration redundancy and concurrency issue.
    - **Performance requirements:** The system is intended to be responsive and accurate.
* **SQL SERVER:**

SQL Server is a relational database management system, or RDBMS, developed and marketed by Microsoft. Similar to other RDBMS software, SQL Server is built on top of SQL, a standard programming language for interacting with the relational databases. SQL server is tied to Transact-SQL, or T-SQL, the Microsoft’s implementation of SQL that adds a set of proprietary programming constructs.SQL Server works exclusively on Windows environment for more than 20 years. In 2016, Microsoft made it available on Linux. SQL Server 2017 became generally available in October 2016 that ran on both Windows and Linux.

As a database server it is software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

* **System Design:** **System Design** is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements.
* **Reliability in System Design -** A system is Reliable when it can meet the end-user requirement. When you are designing a system you should have planned to implement a set of features and services in your system. If your system can serve all those features without wearing out then your System can be considered to be **Reliability in System Design.**
* **Availability**  **in a System Design** - is a characteristic of a System which aims to ensure an agreed level of Operational Performance, also known as **uptime**. It is essential for a system to ensure high availability in order to serve the user’s requests.
* **Scalability in System Design -** refers to the ability of the System to cope up with the increasing load.

**DETAILED DESIGN**

**CHAPTER-3**

**DETAILED DESIGN**

**3.1 Data-flow diagrams(DFD):**

The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and is transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs. For clarity, a key has been provided at the bottom of this page. A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system,where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

**History:**

Larry Constantine, the original developer of structured design, based on Martin and Estrin's "Data Flow Graph" model of computation. Starting in the 1970s, data flow diagrams (DFD) became a popular way to visualize the major steps and data involved in software system processes. DFDs were usually used to show data flow in a computer system, although they could in theory data flows or to explore a new high-level design in terms of data flow.

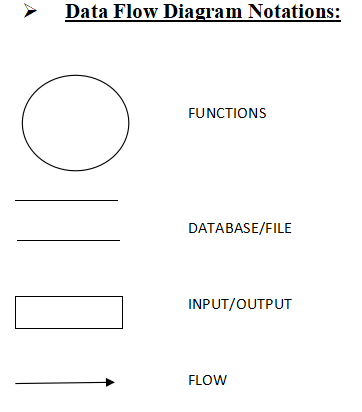
* **Data Flow Diagram Notations:**

FUNCTIONS

DATABASE/FILE

INPUT/OUTPUT

FLOW



* **Physical vs. logical DFD**

A logical DFD captures the data flows that are necessary for a system to operate. It describes the processes that are undertaken, the data required and produced by each process, and the stores needed to hold the data. On the other hand, a physical DFD shows how the system is actually implemented, either at the moment (Current Physical DFD), or how the designer intends it to be in the future (Required Physical DFD). Thus, a Physical DFD may be used to describe the set of data items that appear on each piece of paper that move around an office, and the fact that a particular set of pieces of paper are stored together in a filing cabinet. It is quite possible that a Physical DFD will include references to data that are duplicated, or redundant, and that the data stores, if implemented as a set of database tables, would constitute an UN-normalize (or DE normalized) relational database. In contrast, a Logical DFD attempts to capture the data flow aspects of a system in a form that has neither redundancy nor duplication.

* **DATA FLOW SYMBOLS AND THERE MEANINGS:-**

An entity. A source of data or a destination for data.

**Source/Sink:** Represented by rectangles in the diagram. Sources and Sinks are external entities which are sources or destinations of data, respectively.

**Process:** Represented by circles in the diagram. Processes are responsible for manipulating the data. They take data as input and output an altered version of the data.



**Data Store:** Represented by a segmented rectangle with an open end on the right. Data Stores are both electronic and physical locations of data. Examples include databases, directories, files, and even filing cabinets and stacks of paper.

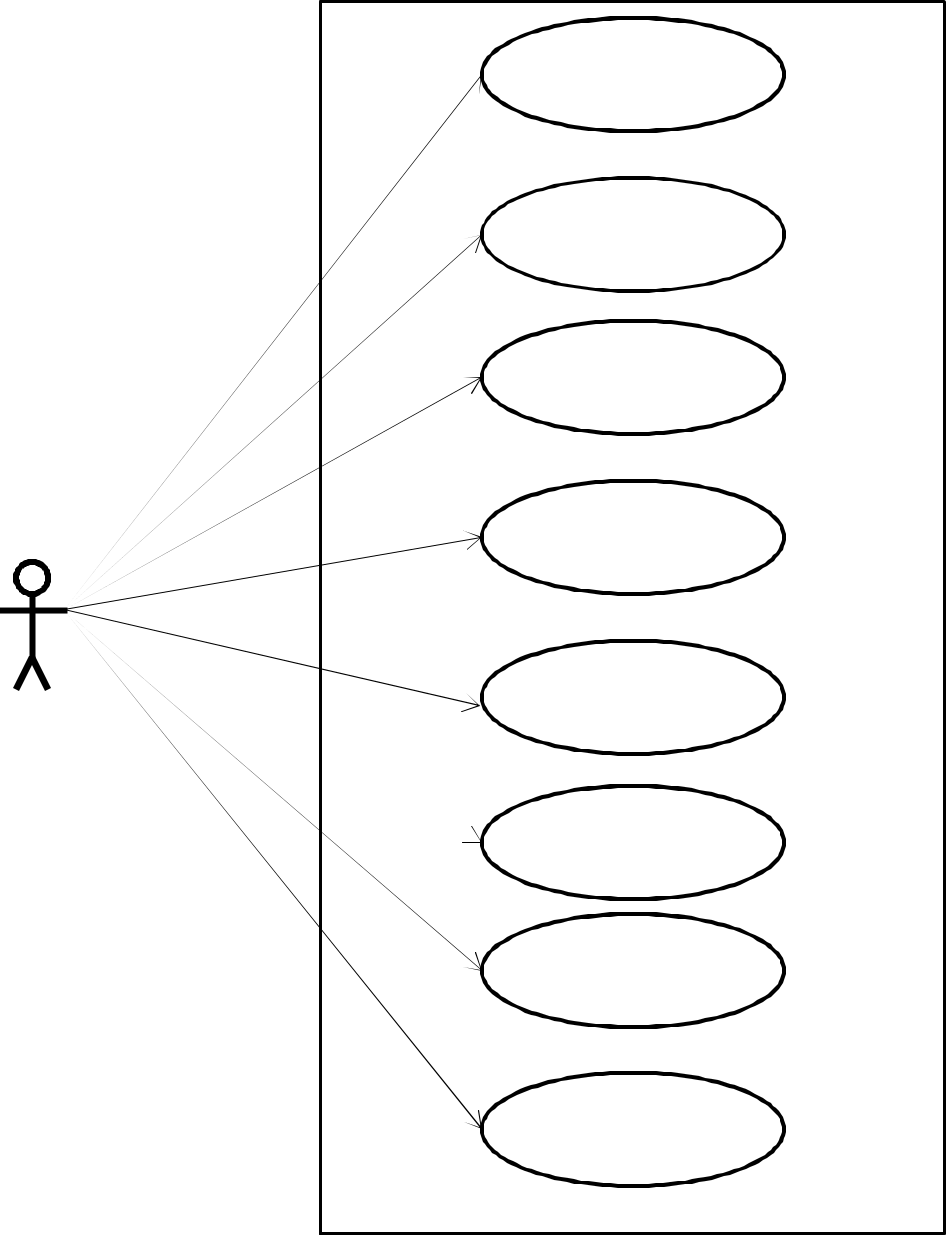
**Data Flow:** Represented by a unidirectional arrow. Data Flows show how data

is moved through the System. Data Flows are labeled with a description of the data that is being passed through it.

In our Application, we need to understand and be able to draw 3 types of Data

Flow Diagrams, they are Level-0, Level 1 and Level 2 DFD’s. A level-0 DFD is the most basic form of DFD. It aims to show how the entire system works at a glance. There is only one process in the system and all the data flows either into or out of this process.Level-0 DFD’s demonstrates the interactions between the process and external entities. They do not contain Data Stores. When drawing Level-0 DFD’s, we must first identify the process, all the external entities and all the data flows. We must also state any assumptions we make about the system. It is advised that we draw the process in the middle of the page. We then draw our external entities in the corners and finally connect our entities to our process with the data flows.

* **DATA FLOW DIAGRAM(LEVEL 0):**
* **DATA FLOW DIAGRAM(ADMIN SIDE):**



Login

Managae user

Manage Loan

Manage Loan Type

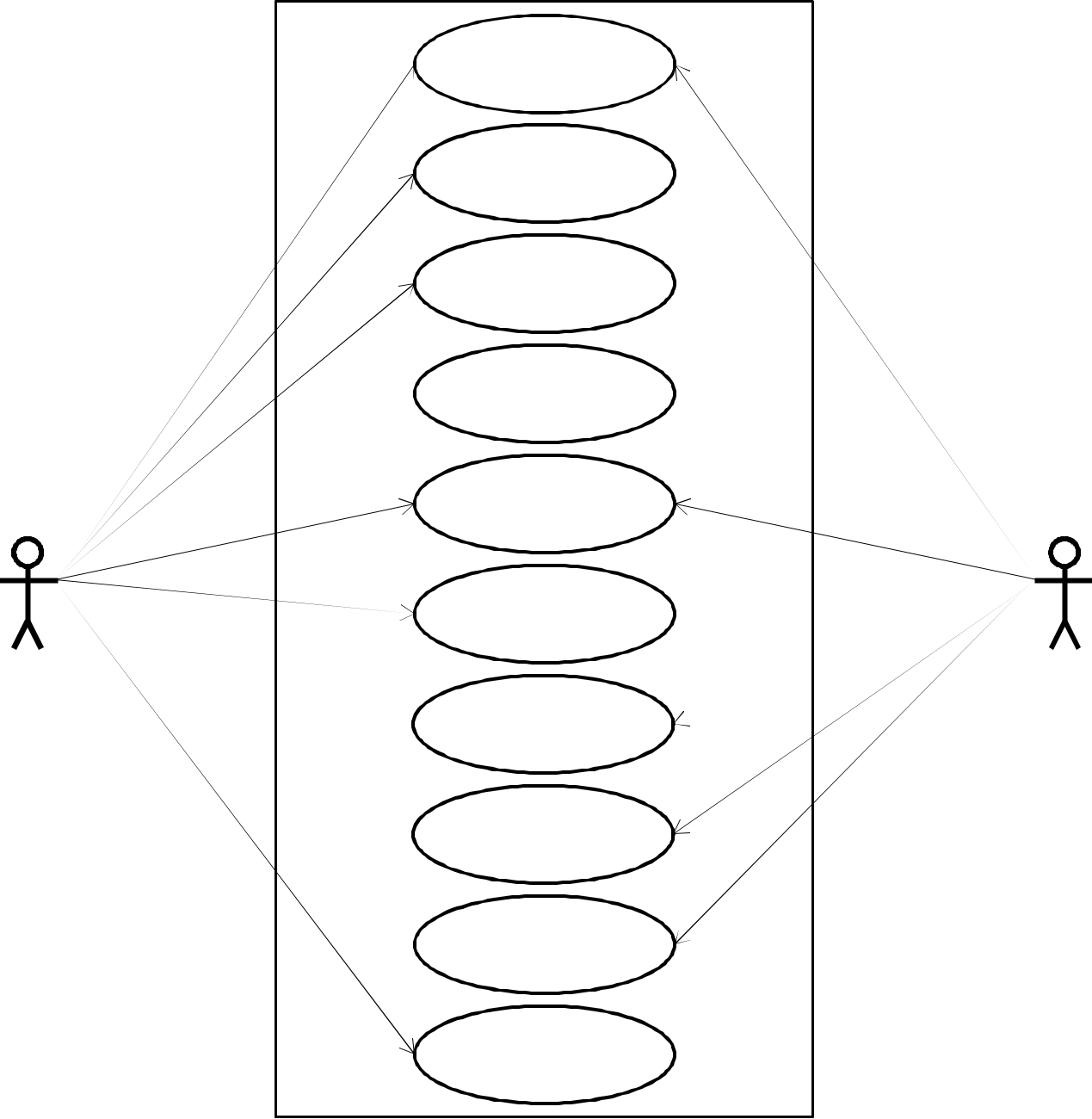
Manage interest rate

Manage EMI Detail

Manage Report

Logout

* **DATA FLOW DIAGRAM(CUSTOMER SIDE)**

****Visit Site

Login

ManageProfile

Manage

Loan Detail

Loan

EMI Detail

Paid EMIDetail

Interest Rate

Detail

Apply online

Loan Form

DownloadForm

Logout

**3.2 Entity Relation.**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique. It Identifies all the relevant entities in a given system and determine the relationships among these entities. An entity should appear only once in a particular diagram. Provide a precise and appropriate name for each entity, attribute, and sounding words. Make effective use of colors. You can use colors to classify similar entities or to highlight key areas in your diagrams. You can draw entity relationship diagrams manually, especially when you are just informally showing simple systems to your peers. However, for more complex systems and for external audiences, you need diagramming software such as Creately’s to craft visually engaging and precise ER diagrams. The ER Diagram Software offered by Create as an online service is pretty easy to use and is a lot more affordable than purchasing licensed software. It is also perfectly suited for development teams because of its strong support for collaboration.

* **The History of Entity Relationship Diagrams:**

Peter Chen developed ERDs in 1976. Since then Charles Bachman and

James Martin have added some slight refinements to the basic ERD principles.

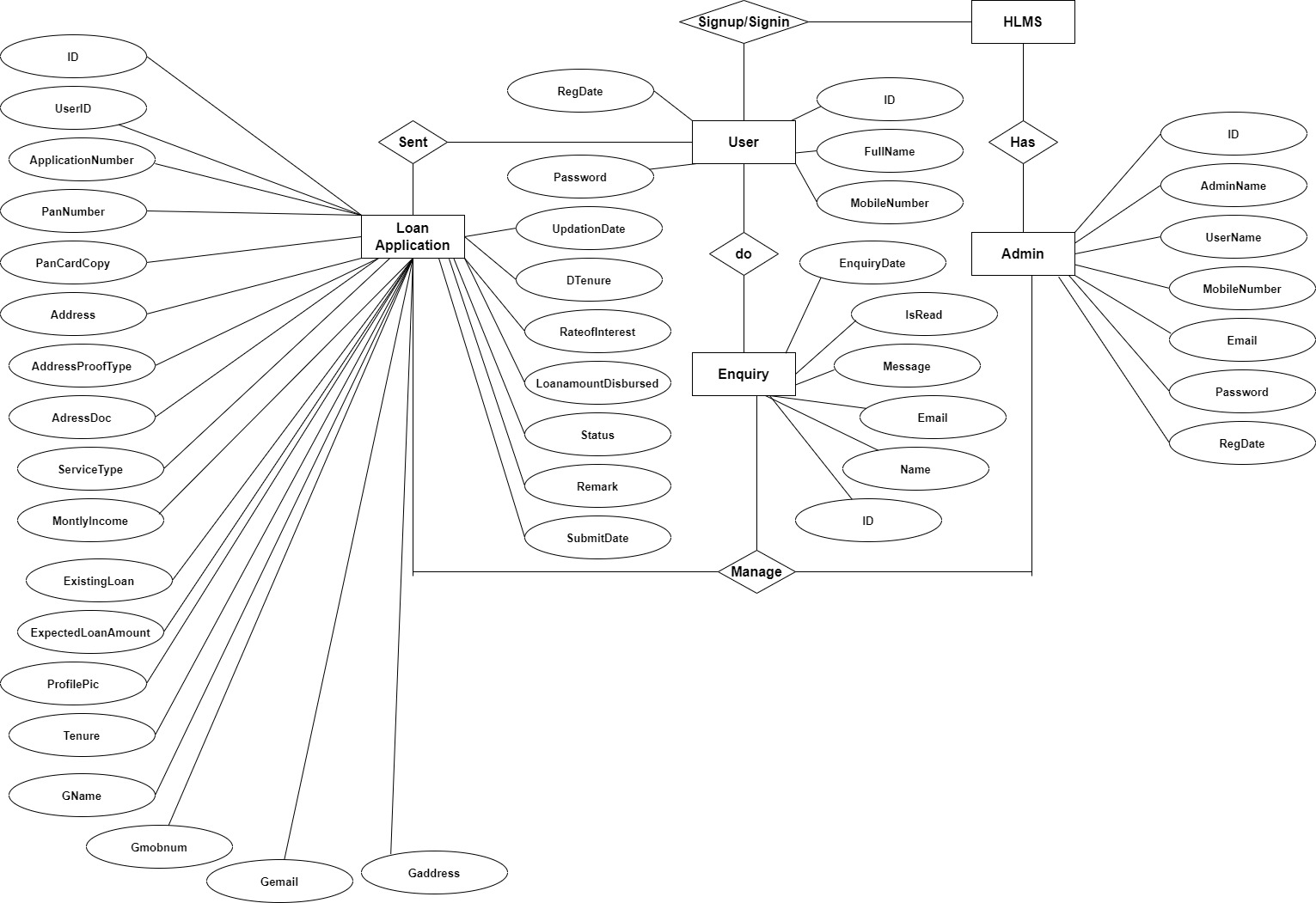
* **Structure of an Entity Relationship Diagram with Common ERD Notations:**

An entity relationship diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

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

Share information in the database. In some cases, entities can be self-linked. For example, Employees can supervise other employees.

* **Attributes** - which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.
* **ER DIAGRAM FOR HOME LOAN:**



**DESIGINING**

**CHAPTER 4**

**DESIGINING**

**4.1 Database design:**

**Database design** is the organization of data according to database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model.Database management system manages the data accordingly.

Database design involves classifying data and identifying interrelationships.

**4.2 Database concepts for opening the section:**

A database is an organized collection of data. Instead of having all the data in a list with a random order, a database provides a structure to organize the data. One of the most common data structures is a database table. A database table consists of rows and columns. A database table is also called a two-dimensional array. An array is like a list of values, and each value is identified by a specific index. A two-dimensional array uses two indices, which correspond to the rows and columns of a table.

In database terminology, each row is called a record. A record is also called an object or an entity. In other words, a database table is a collection of records. The records in a customers in a sales database. A field corresponds to a column in the table and represents a single value for each record. A field is also called an attribute. In other words, a record is a collection of related attributes that make up a single database entry.

**Field(column):** field name uniquely identifies each field. When searching for data in a database, you often specify the field name. Field names for the data in the Instructor file are Instructor ID, First Name, Last Name, Extension, Office, and WebAddress. A database uses a variety of characteristics, such as field size and data type, to define each field.

**Record(Row):** Record is a group of related data held within the same structure. More specifically, a record is a grouping of fields within a table that reference one particular object. The term record is frequently used synonymous with row.

**Table:**A database consists of one or more tables.  Each table is made up of rows and columns.  If you think of a table as a grid, the column go from left to right across the grid and each entry of data is listed down as a row.

Each row in a relational is uniquely  identified by a primary key.  This can be by one or more sets of column values.  In most scenarios it is a single column, such as employee id.

Every relational table has one primary key.  Its purpose is to uniquely identify each row in the database.  No two rows can have the same primary key value.  The practical result of this is that you can select every single row by just knowing its primary key.

**Database:** A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a Database management system(DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

**Relational Database:**A relational database is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

**Primary Key:**A primary key generally focuses on the uniqueness of the table. It is a column or a set of columns that uniquely distinguishes every row in the database. It means it should not have any duplicate value. Also, it doesn’t contain a NULL value.

**Foreign Key:**A foreign key is generally used to build a relationship between the two tables. The major purpose of the foreign key is to sustain data integrity between two separate instances of an entity.

**4.3 Tables:**

A database consists of one or more tables.  Each table is made up of rows and columns.  If you think of a table as a grid, the column go from left to right across the grid and each entry of data is listed down as a row.

Each row in a relational is uniquely  identified by a primary key.  This can be by one or more sets of column values.  In most scenarios it is a single column, such as employee ID.

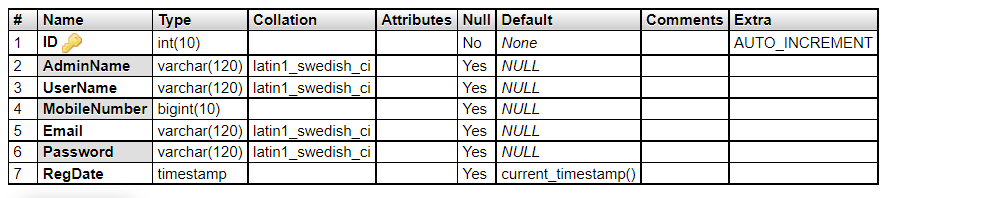
Every relational table has one primary key.  Its purpose is to uniquely identify each row in the database.  No two rows can have the same primary key value.

The practical result of this is that you can select every single row by just knowing its primary key.

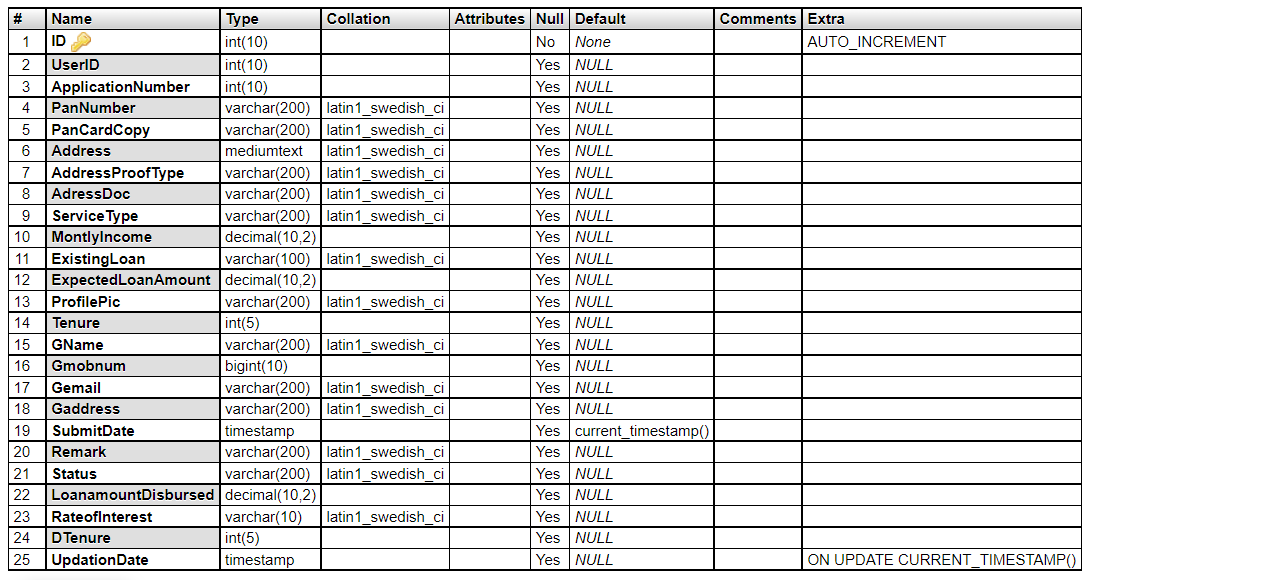
* **CLASS DIAGRAM:**



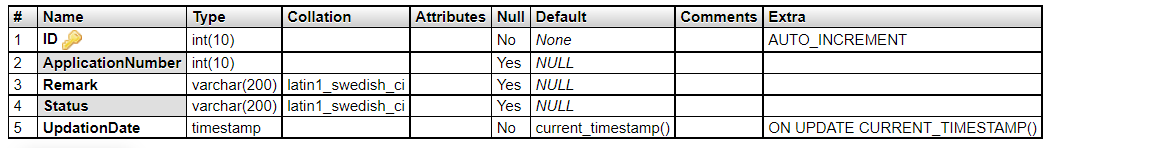
* **TABLE ADMIN:**

****

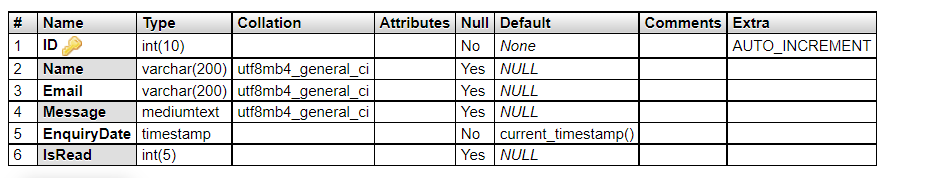
* **CLIENTS TABLE:**

****

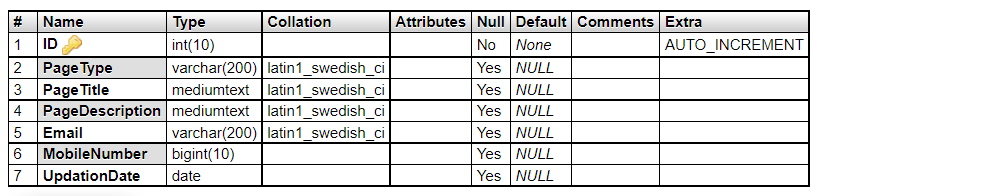
* **APPLICATION TRACKING:**

****

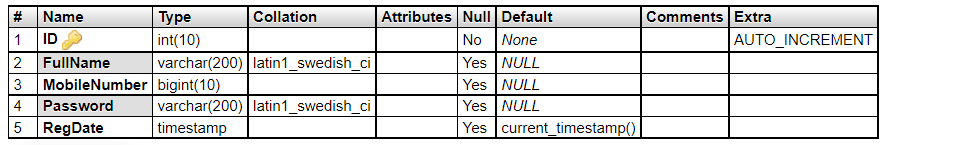
* **TABLE CONTACT:**

****

* **TABLE PAGE:**

****

* **TABLE USER:**

****

**CODING**

**CHAPTER-5**

**CODING**

**5.1 Visual Studio:**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. Visual Studio supports different programming languages and allows the code editor and debugger to support nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, Visual C++ and VB.NET. Support for other languages such as Python, Ruby, Node.js, and M among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Java (and J#) were supported in the past. Microsoft provides a free version of Visual Studio called the Community edition that supports plugins and is available at no cost for all users. Support for programming languages is added by using a specific VS Package called a Language Service. A language service defines various interfaces which the VS Package implementation can implement to add support for various functionalities.

* **Features:**
* Boolean Conditions.
* Automatic Garbage Collection.
* Standard Library.
* Assembly Versioning.
* Properties and Events.
* Delegates and Events Management .
* Easy-to-use Generics .
* Indexers.
* Conditional Compilation.
* Simple Multi -threading.
* **Advantages**

The structure of this Basic programming language is very simple, particularly as to the executable code.

* HTML helps to build structure of a website and is a widely used Markup language.
* It is easy to learn.
* Every browser supports HTML Language.
* HTML is light weighted and fast to load.
* Storage of big files are allowed because of the application cache feature.
* Do not get to purchase any extra software because it’s by default in every window.
* Loose syntax (although, being too flexible won’t suit standards).
* **Disadvantages:**
* It cannot produce dynamic output alone, since it’s a static language.
* Making the structure of HTML documents becomes tough to understand.
* Errors can be costly.
* It is the time consuming as the time it consume to maintain on the color scheme of a page and to make lists, tables and forms.
* We need to write a lot of code for just creating a simple webpage.
* **SQL Server :**

SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called Tabular Data Stream (TDS).

* A Relational DataBase Management System (RDBMS) is software that:
* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

MySQL is a leading open source database management system. It is a multi-user, multi-threaded database management system. MySQL is especially popular on the web. It is one of the parts of the very popular LAMP platform. Linux, Apache, MySQL and PHP. MySQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows or Mac. Wikipedia, YouTube, Facebook use MySQL. These sites manage millions of queries each day. MySQL comes in two versions: MySQL server system and MySQL embedded system.

* **Advantages:-**
* MySQL is becoming so popular because of many good reasons:
* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* **Features** :

**Internals and Portability**

* Written in C and C++.
* Tested with a broad range of different compilers.
* Works on many different platforms.
* For portability, uses CMake in MySQL 5.5 and up. Previous series use GNU Automake, Autoconf, and Libtool.
* Tested with Purify (a commercial memory leakage detector) as well as with Valgrind, a GPL tool.
* **Security:**
* A privilege and password system that is very flexible and secure, and that enables host-based verification.
* Password security by encryption of all password traffic when you connect to a server. Scalability and Limits
* Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
* **Connectivity**
* Clients can connect to MySQL Server using several protocols.
* Clients can connect using TCP/IP sockets on any platform.
* On Unix systems, clients can connect using Unix domain socket files.
* **PHP Hypertext Preprocessor(PHP) :**

**PHP** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning PHP:

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* **HYPERTEXT MARKUP LANGUAGE(HTML):**

**HTML** stands for **Hyper Text Markup Language**, which is the most widely used language on Web to develop web pages. **HTML** was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

* **CASCADING STYLE SHEET(CSS):**

**CSS** is used to control the style of a web document in a simple and easy way.

**CSS** is the acronym for **"Cascading Style Sheet"**. This tutorial covers both the versions CSS1,CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts.

* **JAVASCRIPT(JS):**
* **JavaScript** is a lightweight, interpreted **programming** language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. **JavaScript** is very easy to implement because it is integrated with HTML. It is open and cross-platform.

**5.2 Coding:**

* **INDEX PAGE:**

(function (global, factory) {

typeof exports === 'object' && typeof module !== 'undefined' ? factory(exports, require('jquery'), require('popper.js')) :

typeof define === 'function' && define.amd ? define(['exports', 'jquery', 'popper.js'], factory) :

(factory((global.bootstrap = {}),global.jQuery,global.Popper));

}(this, (function (exports,$,Popper) { 'use strict';

$ = $ && $.hasOwnProperty('default') ? $['default'] : $;

Popper = Popper && Popper.hasOwnProperty('default') ? Popper['default'] : Popper;

function \_defineProperties(target, props) {

for (var i = 0; i < props.length; i++) {

var descriptor = props[i];

descriptor.enumerable = descriptor.enumerable || false;

descriptor.configurable = true;

if ("value" in descriptor) descriptor.writable = true;

Object.defineProperty(target, descriptor.key, descriptor);

}

}

function \_createClass(Constructor, protoProps, staticProps) {

if (protoProps) \_defineProperties(Constructor.prototype, protoProps);

if (staticProps) \_defineProperties(Constructor, staticProps);

return Constructor;

}

function \_extends() {

\_extends = Object.assign || function (target) {

for (var i = 1; i < arguments.length; i++) {

var source = arguments[i];

for (var key in source) {

if (Object.prototype.hasOwnProperty.call(source, key)) {

target[key] = source[key];

}

}

}

return target;

};

return \_extends.apply(this, arguments);

}

function \_inheritsLoose(subClass, superClass) {

subClass.prototype = Object.create(superClass.prototype);

subClass.prototype.constructor = subClass;

subClass.\_\_proto\_\_ = superClass;

}

/\*\*

\* --------------------------------------------------------------------------

\* Bootstrap (v4.0.0): util.js

\* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)

\* --------------------------------------------------------------------------

\*/

var Util = function ($$$1) {

/\*\*

\* ------------------------------------------------------------------------

\* Private TransitionEnd Helpers

\* ------------------------------------------------------------------------

\*/

var transition = false;

var MAX\_UID = 1000000; // Shoutout AngusCroll (https://goo.gl/pxwQGp)

function toType(obj) {

return {}.toString.call(obj).match(/\s([a-zA-Z]+)/)[1].toLowerCase();

}

function getSpecialTransitionEndEvent() {

return {

bindType: transition.end,

delegateType: transition.end,

handle: function handle(event) {

if ($$$1(event.target).is(this)) {

return event.handleObj.handler.apply(this, arguments); // eslint-disable-line prefer-rest-params

}

return undefined; // eslint-disable-line no-undefined

}

};

}

function transitionEndTest() {

if (typeof window !== 'undefined' && window.QUnit) {

return false;

}

return {

end: 'transitionend'

};

}

function transitionEndEmulator(duration) {

var \_this = this;

var called = false;

$$$1(this).one(Util.TRANSITION\_END, function () {

called = true;

});

setTimeout(function () {

if (!called) {

Util.triggerTransitionEnd(\_this);

}

}, duration);

return this;

}

function setTransitionEndSupport() {

transition = transitionEndTest();

$$$1.fn.emulateTransitionEnd = transitionEndEmulator;

if (Util.supportsTransitionEnd()) {

$$$1.event.special[Util.TRANSITION\_END] = getSpecialTransitionEndEvent();

}

}

function escapeId(selector) {

// We escape IDs in case of special selectors (selector = '#myId:something')

// $.escapeSelector does not exist in jQuery < 3

selector = typeof $$$1.escapeSelector === 'function' ? $$$1.escapeSelector(selector).substr(1) : selector.replace(/(:|\.|\[|\]|,|=|@)/g, '\\$1');

return selector;

}

/\*\*

\* --------------------------------------------------------------------------

\* Public Util Api

\* --------------------------------------------------------------------------

\*/

var Util = {

TRANSITION\_END: 'bsTransitionEnd',

getUID: function getUID(prefix) {

do {

// eslint-disable-next-line no-bitwise

prefix += ~~(Math.random() \* MAX\_UID); // "~~" acts like a faster Math.floor() here

} while (document.getElementById(prefix));

return prefix;

},

getSelectorFromElement: function getSelectorFromElement(element) {

var selector = element.getAttribute('data-target');

if (!selector || selector === '#') {

selector = element.getAttribute('href') || '';

} // If it's an ID

if (selector.charAt(0) === '#') {

selector = escapeId(selector);

}

try {

var $selector = $$$1(document).find(selector);

return $selector.length > 0 ? selector : null;

} catch (err) {

return null;

}

},

reflow: function reflow(element) {

return element.offsetHeight;

},

triggerTransitionEnd: function triggerTransitionEnd(element) {

$$$1(element).trigger(transition.end);

},

supportsTransitionEnd: function supportsTransitionEnd() {

return Boolean(transition);

},

isElement: function isElement(obj) {

return (obj[0] || obj).nodeType;

},

typeCheckConfig: function typeCheckConfig(componentName, config, configTypes) {

for (var property in configTypes) {

if (Object.prototype.hasOwnProperty.call(configTypes, property)) {

var expectedTypes = configTypes[property];

var value = config[property];

var valueType = value && Util.isElement(value) ? 'element' : toType(value);

if (!new RegExp(expectedTypes).test(valueType)) {

throw new Error(componentName.toUpperCase() + ": " + ("Option \"" + property + "\" provided type \"" + valueType + "\" ") + ("but expected type \"" + expectedTypes + "\"."));

}

}

}

}

};

setTransitionEndSupport();

return Util;

}($);

/\*\*

\* --------------------------------------------------------------------------

\* Bootstrap (v4.0.0): alert.js

\* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)

\* --------------------------------------------------------------------------

\*/

var Alert = function ($$$1) {

/\*\*

\* ------------------------------------------------------------------------

\* Constants

\* ------------------------------------------------------------------------

\*/

var NAME = 'alert';

var VERSION = '4.0.0';

var DATA\_KEY = 'bs.alert';

var EVENT\_KEY = "." + DATA\_KEY;

var DATA\_API\_KEY = '.data-api';

var JQUERY\_NO\_CONFLICT = $$$1.fn[NAME];

var TRANSITION\_DURATION = 150;

var Selector = {

DISMISS: '[data-dismiss="alert"]'

};

var Event = {

CLOSE: "close" + EVENT\_KEY,

CLOSED: "closed" + EVENT\_KEY,

CLICK\_DATA\_API: "click" + EVENT\_KEY + DATA\_API\_KEY

};

var ClassName = {

ALERT: 'alert',

FADE: 'fade',

SHOW: 'show'

/\*\*

\* ------------------------------------------------------------------------

\* Class Definition

\* ------------------------------------------------------------------------

\*/

};

var Alert =

/\*#\_\_PURE\_\_\*/

function () {

function Alert(element) {

this.\_element = element;

} // Getters

var \_proto = Alert.prototype;

// Public

\_proto.close = function close(element) {

element = element || this.\_element;

var rootElement = this.\_getRootElement(element);

var customEvent = this.\_triggerCloseEvent(rootElement);

if (customEvent.isDefaultPrevented()) {

return;

}

this.\_removeElement(rootElement);

};

\_proto.dispose = function dispose() {

$$$1.removeData(this.\_element, DATA\_KEY);

this.\_element = null;

}; // Private

\_proto.\_getRootElement = function \_getRootElement(element) {

var selector = Util.getSelectorFromElement(element);

var parent = false;

if (selector) {

parent = $$$1(selector)[0];

}

if (!parent) {

parent = $$$1(element).closest("." + ClassName.ALERT)[0];

}

return parent;

};

\_proto.\_triggerCloseEvent = function \_triggerCloseEvent(element) {

var closeEvent = $$$1.Event(Event.CLOSE);

$$$1(element).trigger(closeEvent);

return closeEvent;

};

\_proto.\_removeElement = function \_removeElement(element) {

var \_this = this;

$$$1(element).removeClass(ClassName.SHOW);

if (!Util.supportsTransitionEnd() || !$$$1(element).hasClass(ClassName.FADE)) {

this.\_destroyElement(element);

return;

}

$$$1(element).one(Util.TRANSITION\_END, function (event) {

return \_this.\_destroyElement(element, event);

}).emulateTransitionEnd(TRANSITION\_DURATION);

};

\_proto.\_destroyElement = function \_destroyElement(element) {

$$$1(element).detach().trigger(Event.CLOSED).remove();

}; // Static

Alert.\_jQueryInterface = function \_jQueryInterface(config) {

return this.each(function () {

var $element = $$$1(this);

var data = $element.data(DATA\_KEY);

if (!data) {

data = new Alert(this);

$element.data(DATA\_KEY, data);

}

if (config === 'close') {

data[config](this);

}

});

};

Alert.\_handleDismiss = function \_handleDismiss(alertInstance) {

return function (event) {

if (event) {

event.preventDefault();

}

alertInstance.close(this);

};

};

\_createClass(Alert, null, [{

key: "VERSION",

get: function get() {

return VERSION;

}

}]);

return Alert;

}();

/\*\*

\* ------------------------------------------------------------------------

\* Data Api implementation

\* ------------------------------------------------------------------------

\*/

$$$1(document).on(Event.CLICK\_DATA\_API, Selector.DISMISS, Alert.\_handleDismiss(new Alert()));

/\*\*

\* ------------------------------------------------------------------------

\* jQuery

\* ------------------------------------------------------------------------

\*/

$$$1.fn[NAME] = Alert.\_jQueryInterface;

$$$1.fn[NAME].Constructor = Alert;

$$$1.fn[NAME].noConflict = function () {

$$$1.fn[NAME] = JQUERY\_NO\_CONFLICT;

return Alert.\_jQueryInterface;

};

return Alert;

}($);

/\*\*

\* --------------------------------------------------------------------------

\* Bootstrap (v4.0.0): button.js

\* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)

\* --------------------------------------------------------------------------

\*/

var Button = function ($$$1) {

/\*\*

\* ------------------------------------------------------------------------

\* Constants

\* ------------------------------------------------------------------------

\*/

var NAME = 'button';

var VERSION = '4.0.0';

var DATA\_KEY = 'bs.button';

var EVENT\_KEY = "." + DATA\_KEY;

var DATA\_API\_KEY = '.data-api';

var JQUERY\_NO\_CONFLICT = $$$1.fn[NAME];

var ClassName = {

ACTIVE: 'active',

BUTTON: 'btn',

FOCUS: 'focus'

};

var Selector = {

DATA\_TOGGLE\_CARROT: '[data-toggle^="button"]',

DATA\_TOGGLE: '[data-toggle="buttons"]',

INPUT: 'input',

ACTIVE: '.active',

BUTTON: '.btn'

};

var Event = {

CLICK\_DATA\_API: "click" + EVENT\_KEY + DATA\_API\_KEY,

FOCUS\_BLUR\_DATA\_API: "focus" + EVENT\_KEY + DATA\_API\_KEY + " " + ("blur" + EVENT\_KEY + DATA\_API\_KEY)

/\*\*

\* ------------------------------------------------------------------------

\* Class Definition

\* ------------------------------------------------------------------------

\*/

};

var Button =

/\*#\_\_PURE\_\_\*/

function () {

function Button(element) {

this.\_element = element;

} // Getters

var \_proto = Button.prototype;

// Public

\_proto.toggle = function toggle() {

var triggerChangeEvent = true;

var addAriaPressed = true;

var rootElement = $$$1(this.\_element).closest(Selector.DATA\_TOGGLE)[0];

if (rootElement) {

var input = $$$1(this.\_element).find(Selector.INPUT)[0];

if (input) {

if (input.type === 'radio') {

if (input.checked && $$$1(this.\_element).hasClass(ClassName.ACTIVE)) {

triggerChangeEvent = false;

} else {

var activeElement = $$$1(rootElement).find(Selector.ACTIVE)[0];

if (activeElement) {

$$$1(activeElement).removeClass(ClassName.ACTIVE);

}

}

}

if (triggerChangeEvent) {

if (input.hasAttribute('disabled') || rootElement.hasAttribute('disabled') || input.classList.contains('disabled') || rootElement.classList.contains('disabled')) {

return;

}

input.checked = !$$$1(this.\_element).hasClass(ClassName.ACTIVE);

$$$1(input).trigger('change');

}

input.focus();

addAriaPressed = false;

}

}

if (addAriaPressed) {

this.\_element.setAttribute('aria-pressed', !$$$1(this.\_element).hasClass(ClassName.ACTIVE));

}

if (triggerChangeEvent) {

$$$1(this.\_element).toggleClass(ClassName.ACTIVE);

}

};

\_proto.dispose = function dispose() {

$$$1.removeData(this.\_element, DATA\_KEY);

this.\_element = null;

}; // Static

Button.\_jQueryInterface = function \_jQueryInterface(config) {

return this.each(function () {

var data = $$$1(this).data(DATA\_KEY);

if (!data) {

data = new Button(this);

$$$1(this).data(DATA\_KEY, data);

}

if (config === 'toggle') {

data[config]();

}

});

};

\_createClass(Button, null, [{

key: "VERSION",

get: function get() {

return VERSION;

}

}]);

return Button;

}();

/\*\*

\* ------------------------------------------------------------------------

\* Data Api implementation

\* ------------------------------------------------------------------------

\*/

$$$1(document).on(Event.CLICK\_DATA\_API, Selector.DATA\_TOGGLE\_CARROT, function (event) {

event.preventDefault();

var button = event.target;

if (!$$$1(button).hasClass(ClassName.BUTTON)) {

button = $$$1(button).closest(Selector.BUTTON);

}

Button.\_jQueryInterface.call($$$1(button), 'toggle');

}).on(Event.FOCUS\_BLUR\_DATA\_API, Selector.DATA\_TOGGLE\_CARROT, function (event) {

var button = $$$1(event.target).closest(Selector.BUTTON)[0];

$$$1(button).toggleClass(ClassName.FOCUS, /^focus(in)?$/.test(event.type));

});

/\*\*

\* ------------------------------------------------------------------------

\* jQuery

\* ------------------------------------------------------------------------

\*/

$$$1.fn[NAME] = Button.\_jQueryInterface;

$$$1.fn[NAME].Constructor = Button;

$$$1.fn[NAME].noConflict = function () {

$$$1.fn[NAME] = JQUERY\_NO\_CONFLICT;

return Button.\_jQueryInterface;

};

return Button;

}($);

ss

/\*\*

\* --------------------------------------------------------------------------

\* Bootstrap (v4.0.0): carousel.js

\* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)

\* --------------------------------------------------------------------------

\*/

var Carousel = function ($$$1) {

/\*\*

\* ------------------------------------------------------------------------

\* Constants

\* ------------------------------------------------------------------------

\*/

var NAME = 'carousel';

var VERSION = '4.0.0';

var DATA\_KEY = 'bs.carousel';

var EVENT\_KEY = "." + DATA\_KEY;

var DATA\_API\_KEY = '.data-api';

var JQUERY\_NO\_CONFLICT = $$$1.fn[NAME];

var TRANSITION\_DURATION = 600;

var ARROW\_LEFT\_KEYCODE = 37; // KeyboardEvent.which value for left arrow key

var ARROW\_RIGHT\_KEYCODE = 39; // KeyboardEvent.which value for right arrow key

var TOUCHEVENT\_COMPAT\_WAIT = 500; // Time for mouse compat events to fire after touch

var Default = {

interval: 5000,

keyboard: true,

slide: false,

pause: 'hover',

wrap: true

};

var DefaultType = {

interval: '(number|boolean)',

keyboard: 'boolean',

slide: '(boolean|string)',

pause: '(string|boolean)',

wrap: 'boolean'

};

var Direction = {

NEXT: 'next',

PREV: 'prev',

LEFT: 'left',

RIGHT: 'right'

};

var Event = {

SLIDE: "slide" + EVENT\_KEY,

SLID: "slid" + EVENT\_KEY,

KEYDOWN: "keydown" + EVENT\_KEY,

MOUSEENTER: "mouseenter" + EVENT\_KEY,

MOUSELEAVE: "mouseleave" + EVENT\_KEY,

TOUCHEND: "touchend" + EVENT\_KEY,

LOAD\_DATA\_API: "load" + EVENT\_KEY + DATA\_API\_KEY,

CLICK\_DATA\_API: "click" + EVENT\_KEY + DATA\_API\_KEY

};

var ClassName = {

CAROUSEL: 'carousel',

ACTIVE: 'active',

SLIDE: 'slide',

RIGHT: 'carousel-item-right',

LEFT: 'carousel-item-left',

NEXT: 'carousel-item-next',

PREV: 'carousel-item-prev',

ITEM: 'carousel-item'

};

var Selector = {

ACTIVE: '.active',

ACTIVE\_ITEM: '.active.carousel-item',

ITEM: '.carousel-item',

NEXT\_PREV: '.carousel-item-next, .carousel-item-prev',

INDICATORS: '.carousel-indicators',

DATA\_SLIDE: '[data-slide], [data-slide-to]',

DATA\_RIDE: '[data-ride="carousel"]'

/\*\*

\* ------------------------------------------------------------------------

\* Class Definition

\* ------------------------------------------------------------------------

\*/

};

var Carousel =

/\*#\_\_PURE\_\_\*/

function () {

function Carousel(element, config) {

this.\_items = null;

this.\_interval = null;

this.\_activeElement = null;

this.\_isPaused = false;

this.\_isSliding = false;

this.touchTimeout = null;

this.\_config = this.\_getConfig(config);

this.\_element = $$$1(element)[0];

this.\_indicatorsElement = $$$1(this.\_element).find(Selector.INDICATORS)[0];

this.\_addEventListeners();

} // Getters

var \_proto = Carousel.prototype;

// Public

\_proto.next = function next() {

if (!this.\_isSliding) {

this.\_slide(Direction.NEXT);

}

};

\_proto.nextWhenVisible = function nextWhenVisible() {

// Don't call next when the page isn't visible

// or the carousel or its parent isn't visible

if (!document.hidden && $$$1(this.\_element).is(':visible') && $$$1(this.\_element).css('visibility') !== 'hidden') {

this.next();

}

};

\_proto.prev = function prev() {

if (!this.\_isSliding) {

this.\_slide(Direction.PREV);

}

};

\_proto.pause = function pause(event) {

if (!event) {

this.\_isPaused = true;

}

if ($$$1(this.\_element).find(Selector.NEXT\_PREV)[0] && Util.supportsTransitionEnd()) {

Util.triggerTransitionEnd(this.\_element);

this.cycle(true);

}

clearInterval(this.\_interval);

this.\_interval = null;

};

\_proto.cycle = function cycle(event) {

if (!event) {

this.\_isPaused = false;

}

if (this.\_interval) {

clearInterval(this.\_interval);

this.\_interval = null;

}

if (this.\_config.interval && !this.\_isPaused) {

this.\_interval = setInterval((document.visibilityState ? this.nextWhenVisible : this.next).bind(this), this.\_config.interval);

}

};

\_proto.to = function to(index) {

var \_this = this;

this.\_activeElement = $$$1(this.\_element).find(Selector.ACTIVE\_ITEM)[0];

var activeIndex = this.\_getItemIndex(this.\_activeElement);

if (index > this.\_items.length - 1 || index < 0) {

return;

}

if (this.\_isSliding) {

$$$1(this.\_element).one(Event.SLID, function () {

return \_this.to(index);

});

return;

}

if (activeIndex === index) {

this.pause();

this.cycle();

return;

}

var direction = index > activeIndex ? Direction.NEXT : Direction.PREV;

this.\_slide(direction, this.\_items[index]);

};

\_proto.dispose = function dispose() {

$$$1(this.\_element).off(EVENT\_KEY);

$$$1.removeData(this.\_element, DATA\_KEY);

this.\_items = null;

this.\_config = null;

this.\_element = null;

this.\_interval = null;

this.\_isPaused = null;

this.\_isSliding = null;

this.\_activeElement = null;

this.\_indicatorsElement = null;

}; // Private

\_proto.\_getConfig = function \_getConfig(config) {

config = \_extends({}, Default, config);

Util.typeCheckConfig(NAME, config, DefaultType);

return config;

};

\_proto.\_addEventListeners = function \_addEventListeners() {

var \_this2 = this;

if (this.\_config.keyboard) {

$$$1(this.\_element).on(Event.KEYDOWN, function (event) {

return \_this2.\_keydown(event);

});

}

if (this.\_config.pause === 'hover') {

$$$1(this.\_element).on(Event.MOUSEENTER, function (event) {

return \_this2.pause(event);

}).on(Event.MOUSELEAVE, function (event) {

return \_this2.cycle(event);

});

if ('ontouchstart' in document.documentElement) {

// If it's a touch-enabled device, mouseenter/leave are fired as

// part of the mouse compatibility events on first tap - the carousel

// would stop cycling until user tapped out of it;

// here, we listen for touchend, explicitly pause the carousel

// (as if it's the second time we tap on it, mouseenter compat event

// is NOT fired) and after a timeout (to allow for mouse compatibility

// events to fire) we explicitly restart cycling

$$$1(this.\_element).on(Event.TOUCHEND, function () {

\_this2.pause();

if (\_this2.touchTimeout) {

clearTimeout(\_this2.touchTimeout);

}

\_this2.touchTimeout = setTimeout(function (event) {

return \_this2.cycle(event);

}, TOUCHEVENT\_COMPAT\_WAIT + \_this2.\_config.interval);

});

}

}

};

\_proto.\_keydown = function \_keydown(event) {

if (/input|textarea/i.test(event.target.tagName)) {

return;

}

switch (event.which) {

case ARROW\_LEFT\_KEYCODE:

event.preventDefault();

this.prev();

break;

case ARROW\_RIGHT\_KEYCODE:

event.preventDefault();

this.next();

break;

default:

}

};

\_proto.\_getItemIndex = function \_getItemIndex(element) {

this.\_items = $$$1.makeArray($$$1(element).parent().find(Selector.ITEM));

return this.\_items.indexOf(element);

};

\_proto.show = function show() {

var \_this = this;

if ($$$1(this.element).css('display') === 'none') {

throw new Error('Please use show on visible elements');

}

var showEvent = $$$1.Event(this.constructor.Event.SHOW);

if (this.isWithContent() && this.\_isEnabled) {

$$$1(this.element).trigger(showEvent);

var isInTheDom = $$$1.contains(this.element.ownerDocument.documentElement, this.element);

if (showEvent.isDefaultPrevented() || !isInTheDom) {

return;

}

var tip = this.getTipElement();

var tipId = Util.getUID(this.constructor.NAME);

tip.setAttribute('id', tipId);

this.element.setAttribute('aria-describedby', tipId);

this.setContent();

if (this.config.animation) {

$$$1(tip).addClass(ClassName.FADE);

}

var placement = typeof this.config.placement === 'function' ? this.config.placement.call(this, tip, this.element) : this.config.placement;

var attachment = this.\_getAttachment(placement);

this.addAttachmentClass(attachment);

var container = this.config.container === false ? document.body : $$$1(this.config.container);

$$$1(tip).data(this.constructor.DATA\_KEY, this);

if (!$$$1.contains(this.element.ownerDocument.documentElement, this.tip)) {

$$$1(tip).appendTo(container);

}

$$$1(this.element).trigger(this.constructor.Event.INSERTED);

this.\_popper = new Popper(this.element, tip, {

placement: attachment,

modifiers: {

offset: {

offset: this.config.offset

},

flip: {

behavior: this.config.fallbackPlacement

},

arrow: {

element: Selector.ARROW

},

preventOverflow: {

boundariesElement: this.config.boundary

}

},

onCreate: function onCreate(data) {

if (data.originalPlacement !== data.placement) {

\_this.\_handlePopperPlacementChange(data);

}

},

onUpdate: function onUpdate(data) {

\_this.\_handlePopperPlacementChange(data);

}

});

$$$1(tip).addClass(ClassName.SHOW); // If this is a touch-enabled device we add extra

// empty mouseover listeners to the body's immediate children;

// only needed because of broken event delegation on iOS

// https://www.quirksmode.org/blog/archives/2014/02/mouse\_event\_bub.html

if ('ontouchstart' in document.documentElement) {

$$$1('body').children().on('mouseover', null, $$$1.noop);

}

var complete = function complete() {

if (\_this.config.animation) {

\_this.\_fixTransition();

}

var prevHoverState = \_this.\_hoverState;

\_this.\_hoverState = null;

$$$1(\_this.element).trigger(\_this.constructor.Event.SHOWN);

if (prevHoverState === HoverState.OUT) {

\_this.\_leave(null, \_this);

}

};

if (Util.supportsTransitionEnd() && $$$1(this.tip).hasClass(ClassName.FADE)) {

$$$1(this.tip).one(Util.TRANSITION\_END, complete).emulateTransitionEnd(Tooltip.\_TRANSITION\_DURATION);

} else {

complete();

}

}

};

\_proto.hide = function hide(callback) {

var \_this2 = this;

var tip = this.getTipElement();

var hideEvent = $$$1.Event(this.constructor.Event.HIDE);

var complete = function complete() {

if (\_this2.\_hoverState !== HoverState.SHOW && tip.parentNode) {

tip.parentNode.removeChild(tip);

}

\_this2.\_cleanTipClass();

\_this2.element.removeAttribute('aria-describedby');

$$$1(\_this2.element).trigger(\_this2.constructor.Event.HIDDEN);

if (\_this2.\_popper !== null) {

\_this2.\_popper.destroy();

}

if (callback) {

callback();

}

};

$$$1(this.element).trigger(hideEvent);

if (hideEvent.isDefaultPrevented()) {

return;

}

$$$1(tip).removeClass(ClassName.SHOW); // If this is a touch-enabled device we remove the extra

// empty mouseover listeners we added for iOS support

if ('ontouchstart' in document.documentElement) {

$$$1('body').children().off('mouseover', null, $$$1.noop);

}

this.\_activeTrigger[Trigger.CLICK] = false;

this.\_activeTrigger[Trigger.FOCUS] = false;

this.\_activeTrigger[Trigger.HOVER] = false;

if (Util.supportsTransitionEnd() && $$$1(this.tip).hasClass(ClassName.FADE)) {

$$$1(tip).one(Util.TRANSITION\_END, complete).emulateTransitionEnd(TRANSITION\_DURATION);

} else {

complete();

}

this.\_hoverState = '';

};

\_proto.update = function update() {

if (this.\_popper !== null) {

this.\_popper.scheduleUpdate();

}

}; // Protected

\_proto.isWithContent = function isWithContent() {

return Boolean(this.getTitle());

};

\_proto.addAttachmentClass = function addAttachmentClass(attachment) {

$$$1(this.getTipElement()).addClass(CLASS\_PREFIX + "-" + attachment);

};

\_proto.getTipElement = function getTipElement() {

this.tip = this.tip || $$$1(this.config.template)[0];

return this.tip;

};

\_proto.setContent = function setContent() {

var $tip = $$$1(this.getTipElement());

this.setElementContent($tip.find(Selector.TOOLTIP\_INNER), this.getTitle());

$tip.removeClass(ClassName.FADE + " " + ClassName.SHOW);

};

\_proto.setElementContent = function setElementContent($element, content) {

var html = this.config.html;

if (typeof content === 'object' && (content.nodeType || content.jquery)) {

// Content is a DOM node or a jQuery

if (html) {

if (!$$$1(content).parent().is($element)) {

$element.empty().append(content);

}

} else {

$element.text($$$1(content).text());

}

} else {

$element[html ? 'html' : 'text'](content);

}

};

\_proto.getTitle = function getTitle() {

var title = this.element.getAttribute('data-original-title');

if (!title) {

title = typeof this.config.title === 'function' ? this.config.title.call(this.element) : this.config.title;

}

return title;

}; // Private

\_proto.\_getAttachment = function \_getAttachment(placement) {

return AttachmentMap[placement.toUpperCase()];

};

\_proto.\_setListeners = function \_setListeners() {

var \_this3 = this;

var triggers = this.config.trigger.split(' ');

triggers.forEach(function (trigger) {

if (trigger === 'click') {

$$$1(\_this3.element).on(\_this3.constructor.Event.CLICK, \_this3.config.selector, function (event) {

return \_this3.toggle(event);

});

} else if (trigger !== Trigger.MANUAL) {

var eventIn = trigger === Trigger.HOVER ? \_this3.constructor.Event.MOUSEENTER : \_this3.constructor.Event.FOCUSIN;

var eventOut = trigger === Trigger.HOVER ? \_this3.constructor.Event.MOUSELEAVE : \_this3.constructor.Event.FOCUSOUT;

$$$1(\_this3.element).on(eventIn, \_this3.config.selector, function (event) {

return \_this3.\_enter(event);

}).on(eventOut, \_this3.config.selector, function (event) {

return \_this3.\_leave(event);

});

}

$$$1(\_this3.element).closest('.modal').on('hide.bs.modal', function () {

return \_this3.hide();

});

});

if (this.config.selector) {

this.config = \_extends({}, this.config, {

trigger: 'manual',

selector: ''

});

} else {

this.\_fixTitle();

}

};

\_proto.\_fixTitle = function \_fixTitle() {

var titleType = typeof this.element.getAttribute('data-original-title');

if (this.element.getAttribute('title') || titleType !== 'string') {

this.element.setAttribute('data-original-title', this.element.getAttribute('title') || '');

this.element.setAttribute('title', '');

}

};

\_proto.\_enter = function \_enter(event, context) {

var dataKey = this.constructor.DATA\_KEY;

context = context || $$$1(event.currentTarget).data(dataKey);

if (!context) {

context = new this.constructor(event.currentTarget, this.\_getDelegateConfig());

$$$1(event.currentTarget).data(dataKey, context);

}

if (event) {

context.\_activeTrigger[event.type === 'focusin' ? Trigger.FOCUS : Trigger.HOVER] = true;

}

if ($$$1(context.getTipElement()).hasClass(ClassName.SHOW) || context.\_hoverState === HoverState.SHOW) {

context.\_hoverState = HoverState.SHOW;

return;

}

clearTimeout(context.\_timeout);

context.\_hoverState = HoverState.SHOW;

if (!context.config.delay || !context.config.delay.show) {

context.show();

return;

}

context.\_timeout = setTimeout(function () {

if (context.\_hoverState === HoverState.SHOW) {

context.show();

}

}, context.config.delay.show);

};

\_proto.\_leave = function \_leave(event, context) {

var dataKey = this.constructor.DATA\_KEY;

context = context || $$$1(event.currentTarget).data(dataKey);

if (!context) {

context = new this.constructor(event.currentTarget, this.\_getDelegateConfig());

$$$1(event.currentTarget).data(dataKey, context);

}

if (event) {

context.\_activeTrigger[event.type === 'focusout' ? Trigger.FOCUS : Trigger.HOVER] = false;

}

if (context.\_isWithActiveTrigger()) {

return;

}

clearTimeout(context.\_timeout);

context.\_hoverState = HoverState.OUT;

if (!context.config.delay || !context.config.delay.hide) {

context.hide();

return;

}

context.\_timeout = setTimeout(function () {

if (context.\_hoverState === HoverState.OUT) {

context.hide();

}

}, context.config.delay.hide);

};

\_proto.\_isWithActiveTrigger = function \_isWithActiveTrigger() {

for (var trigger in this.\_activeTrigger) {

if (this.\_activeTrigger[trigger]) {

return true;

}

}

return false;

};

\_proto.\_getConfig = function \_getConfig(config) {

config = \_extends({}, this.constructor.Default, $$$1(this.element).data(), config);

if (typeof config.delay === 'number') {

config.delay = {

show: config.delay,

hide: config.delay

};

}

if (typeof config.title === 'number') {

config.title = config.title.toString();

}

if (typeof config.content === 'number') {

config.content = config.content.toString();

}

Util.typeCheckConfig(NAME, config, this.constructor.DefaultType);

return config;

};

\_proto.\_getDelegateConfig = function \_getDelegateConfig() {

var config = {};

if (this.config) {

for (var key in this.config) {

if (this.constructor.Default[key] !== this.config[key]) {

config[key] = this.config[key];

}

}

}

return config;

};

\_proto.\_cleanTipClass = function \_cleanTipClass() {

var $tip = $$$1(this.getTipElement());

var tabClass = $tip.attr('class').match(BSCLS\_PREFIX\_REGEX);

if (tabClass !== null && tabClass.length > 0) {

$tip.removeClass(tabClass.join(''));

}

};

\_proto.\_handlePopperPlacementChange = function \_handlePopperPlacementChange(data) {

this.\_cleanTipClass();

this.addAttachmentClass(this.\_getAttachment(data.placement));

};

\_proto.\_fixTransition = function \_fixTransition() {

var tip = this.getTipElement();

var initConfigAnimation = this.config.animation;

if (tip.getAttribute('x-placement') !== null) {

return;

}

$$$1(tip).removeClass(ClassName.FADE);

this.config.animation = false;

this.hide();

this.show();

this.config.animation = initConfigAnimation;

}; // Static

Tooltip.\_jQueryInterface = function \_jQueryInterface(config) {

return this.each(function () {

var data = $$$1(this).data(DATA\_KEY);

var \_config = typeof config === 'object' && config;

if (!data && /dispose|hide/.test(config)) {

return;

}

if (!data) {

data = new Tooltip(this, \_config);

$$$1(this).data(DATA\_KEY, data);

}

if (typeof config === 'string') {

if (typeof data[config] === 'undefined') {

throw new TypeError("No method named \"" + config + "\"");

}

data[config]();

}

});

};

\_createClass(Tooltip, null, [{

key: "VERSION",

get: function get() {

return VERSION;

}

}, {

key: "Default",

get: function get() {

return Default;

}

}, {

key: "NAME",

get: function get() {

return NAME;

}

}, {

key: "DATA\_KEY",

get: function get() {

return DATA\_KEY;

}

}, {

key: "Event",

get: function get() {

return Event;

}

}, {

key: "EVENT\_KEY",

get: function get() {

return EVENT\_KEY;

}

}, {

key: "DefaultType",

get: function get() {

return DefaultType;

}

}]);

return Tooltip;

}();

/\*\*

\* ------------------------------------------------------------------------

\* jQuery

\* ------------------------------------------------------------------------

\*/

$$$1.fn[NAME] = Tooltip.\_jQueryInterface;

$$$1.fn[NAME].Constructor = Tooltip;

$$$1.fn[NAME].noConflict = function () {

$$$1.fn[NAME] = JQUERY\_NO\_CONFLICT;

return Tooltip.\_jQueryInterface;

};

return Tooltip;

}($, Popper);

/\*\*

\* --------------------------------------------------------------------------

\* Bootstrap (v4.0.0): popover.js

\* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)

\* --------------------------------------------------------------------------

\*/

var Popover = function ($$$1) {

/\*\*

\* ------------------------------------------------------------------------

\* Constants

\* ------------------------------------------------------------------------

\*/

var NAME = 'popover';

var VERSION = '4.0.0';

var DATA\_KEY = 'bs.popover';

var EVENT\_KEY = "." + DATA\_KEY;

var JQUERY\_NO\_CONFLICT = $$$1.fn[NAME];

var CLASS\_PREFIX = 'bs-popover';

var BSCLS\_PREFIX\_REGEX = new RegExp("(^|\\s)" + CLASS\_PREFIX + "\\S+", 'g');

var Default = \_extends({}, Tooltip.Default, {

placement: 'right',

trigger: 'click',

content: '',

template: '<div class="popover" role="tooltip">' + '<div class="arrow"></div>' + '<h3 class="popover-header"></h3>' + '<div class="popover-body"></div></div>'

});

var DefaultType = \_extends({}, Tooltip.DefaultType, {

content: '(string|element|function)'

});

var ClassName = {

FADE: 'fade',

SHOW: 'show'

};

var Selector = {

TITLE: '.popover-header',

CONTENT: '.popover-body'

};

var Event = {

HIDE: "hide" + EVENT\_KEY,

HIDDEN: "hidden" + EVENT\_KEY,

SHOW: "show" + EVENT\_KEY,

SHOWN: "shown" + EVENT\_KEY,

INSERTED: "inserted" + EVENT\_KEY,

CLICK: "click" + EVENT\_KEY,

FOCUSIN: "focusin" + EVENT\_KEY,

FOCUSOUT: "focusout" + EVENT\_KEY,

MOUSEENTER: "mouseenter" + EVENT\_KEY,

MOUSELEAVE: "mouseleave" + EVENT\_KEY

/\*\*

\* ------------------------------------------------------------------------

\* Class Definition

\* ------------------------------------------------------------------------

\*/

};

var Popover =

/\*#\_\_PURE\_\_\*/

function (\_Tooltip) {

\_inheritsLoose(Popover, \_Tooltip);

function Popover() {

return \_Tooltip.apply(this, arguments) || this;

}

var \_proto = Popover.prototype;

// Overrides

\_proto.isWithContent = function isWithContent() {

return this.getTitle() || this.\_getContent();

};

\_proto.addAttachmentClass = function addAttachmentClass(attachment) {

$$$1(this.getTipElement()).addClass(CLASS\_PREFIX + "-" + attachment);

};

\_proto.getTipElement = function getTipElement() {

this.tip = this.tip || $$$1(this.config.template)[0];

return this.tip;

};

\_proto.setContent = function setContent() {

var $tip = $$$1(this.getTipElement()); // We use append for html objects to maintain js events

this.setElementContent($tip.find(Selector.TITLE), this.getTitle());

var content = this.\_getContent();

if (typeof content === 'function') {

content = content.call(this.element);

}

this.setElementContent($tip.find(Selector.CONTENT), content);

$tip.removeClass(ClassName.FADE + " " + ClassName.SHOW);

}; // Private

\_proto.\_getContent = function \_getContent() {

return this.element.getAttribute('data-content') || this.config.content;

};

\_proto.\_cleanTipClass = function \_cleanTipClass() {

var $tip = $$$1(this.getTipElement());

var tabClass = $tip.attr('class').match(BSCLS\_PREFIX\_REGEX);

if (tabClass !== null && tabClass.length > 0) {

$tip.removeClass(tabClass.join(''));

}

}; // Static

Popover.\_jQueryInterface = function \_jQueryInterface(config) {

return this.each(function () {

var data = $$$1(this).data(DATA\_KEY);

var \_config = typeof config === 'object' ? config : null;

if (!data && /destroy|hide/.test(config)) {

return;

}

if (!data) {

data = new Popover(this, \_config);

$$$1(this).data(DATA\_KEY, data);

}

if (typeof config === 'string') {

if (typeof data[config] === 'undefined') {

throw new TypeError("No method named \"" + config + "\"");

}

data[config]();

}

});

};

\_createClass(Popover, null, [{

key: "VERSION",

// Getters

get: function get() {

return VERSION;

}

}, {

key: "Default",

get: function get() {

return Default;

}

}, {

key: "NAME",

get: function get() {

return NAME;

}

}, {

key: "DATA\_KEY",

get: function get() {

return DATA\_KEY;

}

}, {

key: "Event",

get: function get() {

return Event;

}

}, {

key: "EVENT\_KEY",

get: function get() {

return EVENT\_KEY;

}

}, {

key: "DefaultType",

get: function get() {

return DefaultType;

}

}]);

return Popover;

}(Tooltip);

/\*\*

\* ------------------------------------------------------------------------

\* jQuery

\* ------------------------------------------------------------------------

\*/

$$$1.fn[NAME] = Popover.\_jQueryInterface;

$$$1.fn[NAME].Constructor = Popover;

$$$1.fn[NAME].noConflict = function () {

$$$1.fn[NAME] = JQUERY\_NO\_CONFLICT;

return Popover.\_jQueryInterface;

};

return Popover;

}($);

/\*\*

\* --------------------------------------------------------------------------

\* Bootstrap (v4.0.0): scrollspy.js

\* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)

\* --------------------------------------------------------------------------

\*/

var ScrollSpy = function ($$$1) {

/\*\*

\* ------------------------------------------------------------------------

\* Constants

\* ------------------------------------------------------------------------

\*/

var NAME = 'scrollspy';

var VERSION = '4.0.0';

var DATA\_KEY = 'bs.scrollspy';

var EVENT\_KEY = "." + DATA\_KEY;

var DATA\_API\_KEY = '.data-api';

var JQUERY\_NO\_CONFLICT = $$$1.fn[NAME];

var Default = {

offset: 10,

method: 'auto',

target: ''

};

var DefaultType = {

offset: 'number',

method: 'string',

target: '(string|element)'

};

var Event = {

ACTIVATE: "activate" + EVENT\_KEY,

SCROLL: "scroll" + EVENT\_KEY,

LOAD\_DATA\_API: "load" + EVENT\_KEY + DATA\_API\_KEY

};

var ClassName = {

DROPDOWN\_ITEM: 'dropdown-item',

DROPDOWN\_MENU: 'dropdown-menu',

ACTIVE: 'active'

};

var Selector = {

DATA\_SPY: '[data-spy="scroll"]',

ACTIVE: '.active',

NAV\_LIST\_GROUP: '.nav, .list-group',

NAV\_LINKS: '.nav-link',

NAV\_ITEMS: '.nav-item',

LIST\_ITEMS: '.list-group-item',

DROPDOWN: '.dropdown',

DROPDOWN\_ITEMS: '.dropdown-item',

DROPDOWN\_TOGGLE: '.dropdown-toggle'

};

var OffsetMethod = {

OFFSET: 'offset',

POSITION: 'position'

/\*\*

\* ------------------------------------------------------------------------

\* Class Definition

\* ------------------------------------------------------------------------

\*/

};

var ScrollSpy =

/\*#\_\_PURE\_\_\*/

function () {

function ScrollSpy(element, config) {

var \_this = this;

this.\_element = element;

this.\_scrollElement = element.tagName === 'BODY' ? window : element;

this.\_config = this.\_getConfig(config);

this.\_selector = this.\_config.target + " " + Selector.NAV\_LINKS + "," + (this.\_config.target + " " + Selector.LIST\_ITEMS + ",") + (this.\_config.target + " " + Selector.DROPDOWN\_ITEMS);

this.\_offsets = [];

this.\_targets = [];

this.\_activeTarget = null;

this.\_scrollHeight = 0;

$$$1(this.\_scrollElement).on(Event.SCROLL, function (event) {

return \_this.\_process(event);

});

this.refresh();

this.\_process();

} // Getters

var \_proto = ScrollSpy.prototype;

// Public

\_proto.refresh = function refresh() {

var \_this2 = this;

var autoMethod = this.\_scrollElement === this.\_scrollElement.window ? OffsetMethod.OFFSET : OffsetMethod.POSITION;

var offsetMethod = this.\_config.method === 'auto' ? autoMethod : this.\_config.method;

var offsetBase = offsetMethod === OffsetMethod.POSITION ? this.\_getScrollTop() : 0;

this.\_offsets = [];

this.\_targets = [];

this.\_scrollHeight = this.\_getScrollHeight();

var targets = $$$1.makeArray($$$1(this.\_selector));

targets.map(function (element) {

var target;

var targetSelector = Util.getSelectorFromElement(element);

if (targetSelector) {

target = $$$1(targetSelector)[0];

}

if (target) {

var targetBCR = target.getBoundingClientRect();

if (targetBCR.width || targetBCR.height) {

// TODO (fat): remove sketch reliance on jQuery position/offset

return [$$$1(target)[offsetMethod]().top + offsetBase, targetSelector];

}

}

return null;

}).filter(function (item) {

return item;

}).sort(function (a, b) {

return a[0] - b[0];

}).forEach(function (item) {

\_this2.\_offsets.push(item[0]);

\_this2.\_targets.push(item[1]);

});

};

\_proto.dispose = function dispose() {

$$$1.removeData(this.\_element, DATA\_KEY);

$$$1(this.\_scrollElement).off(EVENT\_KEY);

this.\_element = null;

this.\_scrollElement = null;

this.\_config = null;

this.\_selector = null;

this.\_offsets = null;

this.\_targets = null;

this.\_activeTarget = null;

this.\_scrollHeight = null;

}; // Private

\_proto.\_getConfig = function \_getConfig(config) {

config = \_extends({}, Default, config);

if (typeof config.target !== 'string') {

var id = $$$1(config.target).attr('id');

if (!id) {

id = Util.getUID(NAME);

$$$1(config.target).attr('id', id);

}

config.target = "#" + id;

}

Util.typeCheckConfig(NAME, config, DefaultType);

return config;

};

\_proto.\_getScrollTop = function \_getScrollTop() {

return this.\_scrollElement === window ? this.\_scrollElement.pageYOffset : this.\_scrollElement.scrollTop;

};

\_proto.\_getScrollHeight = function \_getScrollHeight() {

return this.\_scrollElement.scrollHeight || Math.max(document.body.scrollHeight, document.documentElement.scrollHeight);

};

\_proto.\_getOffsetHeight = function \_getOffsetHeight() {

return this.\_scrollElement === window ? window.innerHeight : this.\_scrollElement.getBoundingClientRect().height;

};

\_proto.\_process = function \_process() {

var scrollTop = this.\_getScrollTop() + this.\_config.offset;

var scrollHeight = this.\_getScrollHeight();

var maxScroll = this.\_config.offset + scrollHeight - this.\_getOffsetHeight();

if (this.\_scrollHeight !== scrollHeight) {

this.refresh();

}

if (scrollTop >= maxScroll) {

var target = this.\_targets[this.\_targets.length - 1];

if (this.\_activeTarget !== target) {

this.\_activate(target);

}

return;

}

if (this.\_activeTarget && scrollTop < this.\_offsets[0] && this.\_offsets[0] > 0) {

this.\_activeTarget = null;

this.\_clear();

return;

}

for (var i = this.\_offsets.length; i--;) {

var isActiveTarget = this.\_activeTarget !== this.\_targets[i] && scrollTop >= this.\_offsets[i] && (typeof this.\_offsets[i + 1] === 'undefined' || scrollTop < this.\_offsets[i + 1]);

if (isActiveTarget) {

this.\_activate(this.\_targets[i]);

}

}

};

\_proto.\_activate = function \_activate(target) {

this.\_activeTarget = target;

this.\_clear();

var queries = this.\_selector.split(','); // eslint-disable-next-line arrow-body-style

queries = queries.map(function (selector) {

return selector + "[data-target=\"" + target + "\"]," + (selector + "[href=\"" + target + "\"]");

});

var $link = $$$1(queries.join(','));

if ($link.hasClass(ClassName.DROPDOWN\_ITEM)) {

$link.closest(Selector.DROPDOWN).find(Selector.DROPDOWN\_TOGGLE).addClass(ClassName.ACTIVE);

$link.addClass(ClassName.ACTIVE);

} else {

// Set triggered link as active

$link.addClass(ClassName.ACTIVE); // Set triggered links parents as active

// With both <ul> and <nav> markup a parent is the previous sibling of any nav ancestor

$link.parents(Selector.NAV\_LIST\_GROUP).prev(Selector.NAV\_LINKS + ", " + Selector.LIST\_ITEMS).addClass(ClassName.ACTIVE); // Handle special case when .nav-link is inside .nav-item

$link.parents(Selector.NAV\_LIST\_GROUP).prev(Selector.NAV\_ITEMS).children(Selector.NAV\_LINKS).addClass(ClassName.ACTIVE);

}

$$$1(this.\_scrollElement).trigger(Event.ACTIVATE, {

relatedTarget: target

});

};

\_proto.\_clear = function \_clear() {

$$$1(this.\_selector).filter(Selector.ACTIVE).removeClass(ClassName.ACTIVE);

}; // Static

ScrollSpy.\_jQueryInterface = function \_jQueryInterface(config) {

return this.each(function () {

var data = $$$1(this).data(DATA\_KEY);

var \_config = typeof config === 'object' && config;

if (!data) {

data = new ScrollSpy(this, \_config);

$$$1(this).data(DATA\_KEY, data);

}

if (typeof config === 'string') {

if (typeof data[config] === 'undefined') {

throw new TypeError("No method named \"" + config + "\"");

}

data[config]();

}

});

};

\_createClass(ScrollSpy, null, [{

key: "VERSION",

get: function get() {

return VERSION;

}

}, {

key: "Default",

get: function get() {

return Default;

}

}]);

return ScrollSpy;

}();

/\*\*

\* ------------------------------------------------------------------------

\* Data Api implementation

\* ------------------------------------------------------------------------

\*/

$$$1(window).on(Event.LOAD\_DATA\_API, function () {

var scrollSpys = $$$1.makeArray($$$1(Selector.DATA\_SPY));

for (var i = scrollSpys.length; i--;) {

var $spy = $$$1(scrollSpys[i]);

ScrollSpy.\_jQueryInterface.call($spy, $spy.data());

}

});

/\*\*

\* ------------------------------------------------------------------------

\* jQuery

\* ------------------------------------------------------------------------

\*/

$$$1.fn[NAME] = ScrollSpy.\_jQueryInterface;

$$$1.fn[NAME].Constructor = ScrollSpy;

$$$1.fn[NAME].noConflict = function () {

$$$1.fn[NAME] = JQUERY\_NO\_CONFLICT;

return ScrollSpy.\_jQueryInterface;

};

<!DOCTYPE html>

<html lang="en">

<head>

<title>Home Loan Maanagement System|| Profile</title>

<!-- base:css -->

<link rel="stylesheet" href="vendors/mdi/css/materialdesignicons.min.css">

<link rel="stylesheet" href="vendors/feather/feather.css">

<link rel="stylesheet" href="vendors/base/vendor.bundle.base.css">

<!-- endinject -->

<!-- plugin css for this page -->

<link rel="stylesheet" href="vendors/select2/select2.min.css">

<link rel="stylesheet" href="vendors/select2-bootstrap-theme/select2-bootstrap.min.css">

<!-- End plugin css for this page -->

<!-- inject:css -->

<link rel="stylesheet" href="css/style.css">

<!-- endinject -->

<link rel="shortcut icon" href="images/favicon.png" />

</head>

<body>

<div class="container-scroller">

<?php include\_once('includes/header.php');?>

<div class="container-fluid page-body-wrapper">

<?php include\_once('includes/sidebar.php');?>

<div class="main-panel">

<div class="content-wrapper">

<div class="row">

<div class="col-12 grid-margin stretch-card">

<div class="card">

<div class="card-body">

<h3>Admin Profile</h3>

<br>

<form class="forms-sample" method="post">

<?php

$sql="SELECT \* from tbladmin";

$query = $dbh -> prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$cnt=1;

if($query->rowCount() > 0)

{

foreach($results as $row)

{ ?>

<div class="form-group">

<label for="exampleInputName1">Admin Name</label>

<input type="text" class="form-control" id="exampleTextInput1" name="adminname" value="<?php echo $row->AdminName;?>" required='true'>

</div>

<div class="form-group">

<label for="exampleInputEmail3">User Name</label>

<input type="text" class="form-control" id="email2" name="username" value="<?php echo $row->UserName;?>" readonly="true">

</div>

<div class="form-group">

<label for="exampleInputPassword4">Email</label>

<input type="email" class="form-control" id="email2" name="email" value="<?php echo $row->Email;?>" required='true'>

</div>

<div class="form-group">

<label for="exampleSelectGender">Contact Number</label>

<input type="text" class="form-control" id="email2" name="mobilenumber" value="<?php echo $row->MobileNumber;?>" required='true' maxlength='10'>

</div>

<div class="form-group">

<label for="exampleSelectGender">Registration Date</label>

<input type="text" class="form-control" id="email2" name="" value="<?php echo $row->RegDate;?>" readonly="true">

</div>

<?php $cnt=$cnt+1;}} ?>

<br>

<button type="submit" class="btn btn-primary mr-2" name="submit">Update</button>

</form>

</div>

</div>

</div>

</div>

</div>

<!-- content-wrapper ends -->

<?php include\_once('includes/footer.php');?>

</div>

<!-- main-panel ends -->

</div>

<!-- page-body-wrapper ends -->

</div>

<!-- container-scroller -->

<!-- base:js -->

<script src="vendors/base/vendor.bundle.base.js"></script>

<!-- endinject -->

<!-- inject:js -->

<script src="js/off-canvas.js"></script>

<script src="js/hoverable-collapse.js"></script>

<script src="js/template.js"></script>

<!-- endinject -->

<!-- plugin js for this page -->

<script src="vendors/typeahead.js/typeahead.bundle.min.js"></script>

<script src="vendors/select2/select2.min.js"></script>

<!-- End plugin js for this page -->

<!-- Custom js for this page-->

<script src="js/file-upload.js"></script>

<script src="js/typeahead.js"></script>

<script src="js/select2.js"></script>

<!-- End custom js for this page-->

</body>

</html>

<?php } ?>

**Testing**

**CHAPTER-6**

**Testing**

**6.1 Introduction:**

**Testing:**

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

Testing is a vital part of software development, and it is important to start it as early as possible, and to make testing a part of the process of deciding requirements. To get the most useful perspective on your development project, it is worthwhile devoting some thought to the entire life cycle including how feedback from users will influence the future of the application. The tools and techniques we've discussed in this book should help your team to be more responsive to changes without extra cost, despite the necessarily wide variety of different development processes. Nevertheless, new tools and process improvements should be adopted gradually, assessing the results after each step.

Testing is part of a life cycle. The software development life cycle is one in which you hear of a need, you write some code to fulfil it, and then you check to see whether you have pleased the stakeholders—the users, owners, and other people who have an interest in what the software does. Hopefully they like it, but would also like some additions or changes, so you update or augment your code; and so the cycle continues.

.

* **Software Testing Types:**
* **Black box testing–** Internal system design is not considered in this type of testing. Tests are based on requirements and functionality. White box testing – This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.
* **Unit testing –** Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.
* **Incremental integration testing** – Bottom up approach for testing i.e continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. done by programmers or by testers.
* **Integration testing** – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.
* **Functional testing**–This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.
* **System testing –** Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.
* **End-to-end testing -**  Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.
* **Sanity testing** – Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use then system is not stable enough for further testing and build or application is assigned to fix.

**6.3 System Implementation:**

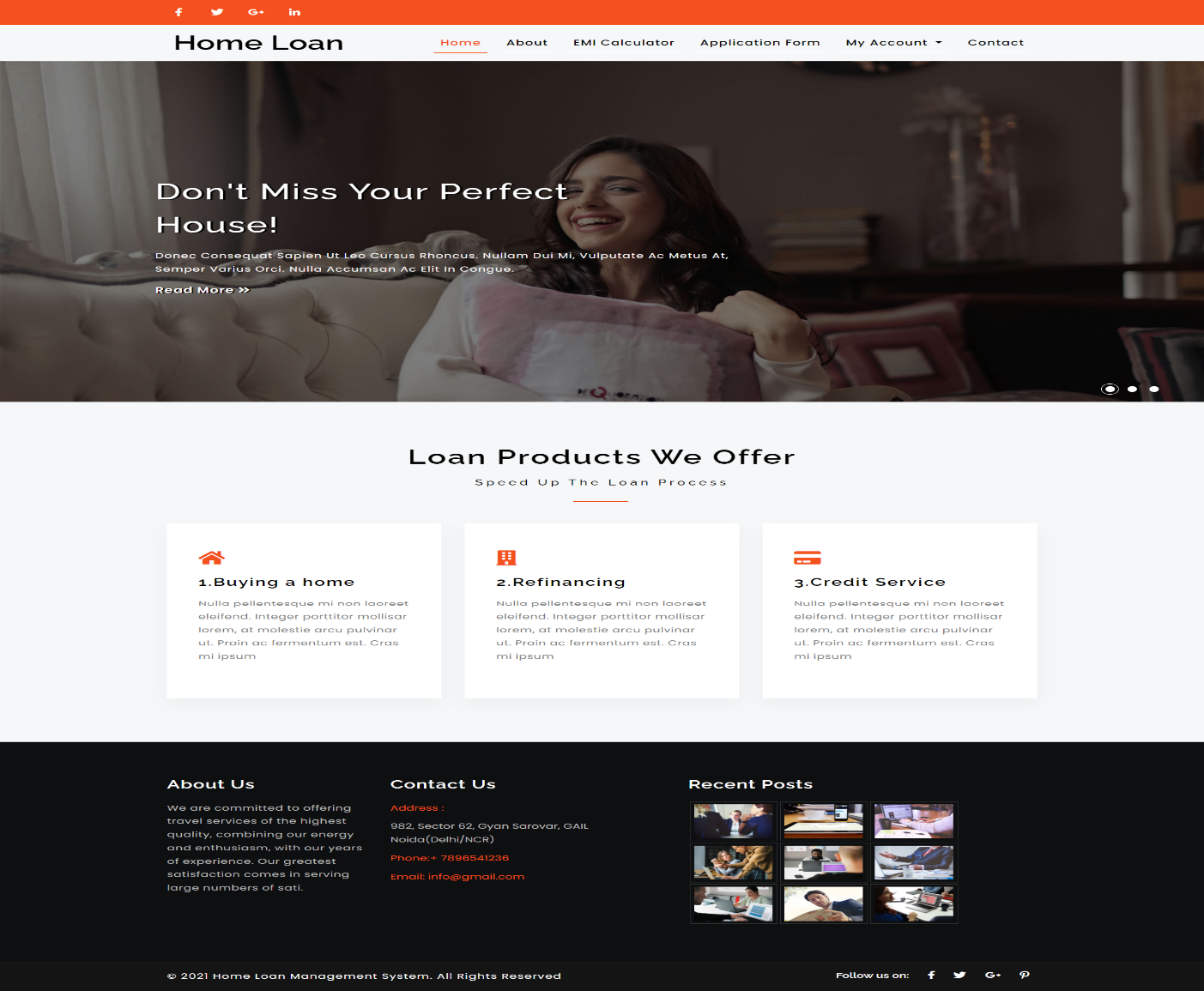
Implementation is the stage of project when the theoretical design is turned into working system.Thus it can be considered to be the critical stage in achieving a successful new system and in giving the user,confidence that system will work and be effective.The implementation stage involves careful planning ,investigation of existing System and its constraints on implementation ,designing of method to achieve changeover and changeover methods.

**Screen Layout**

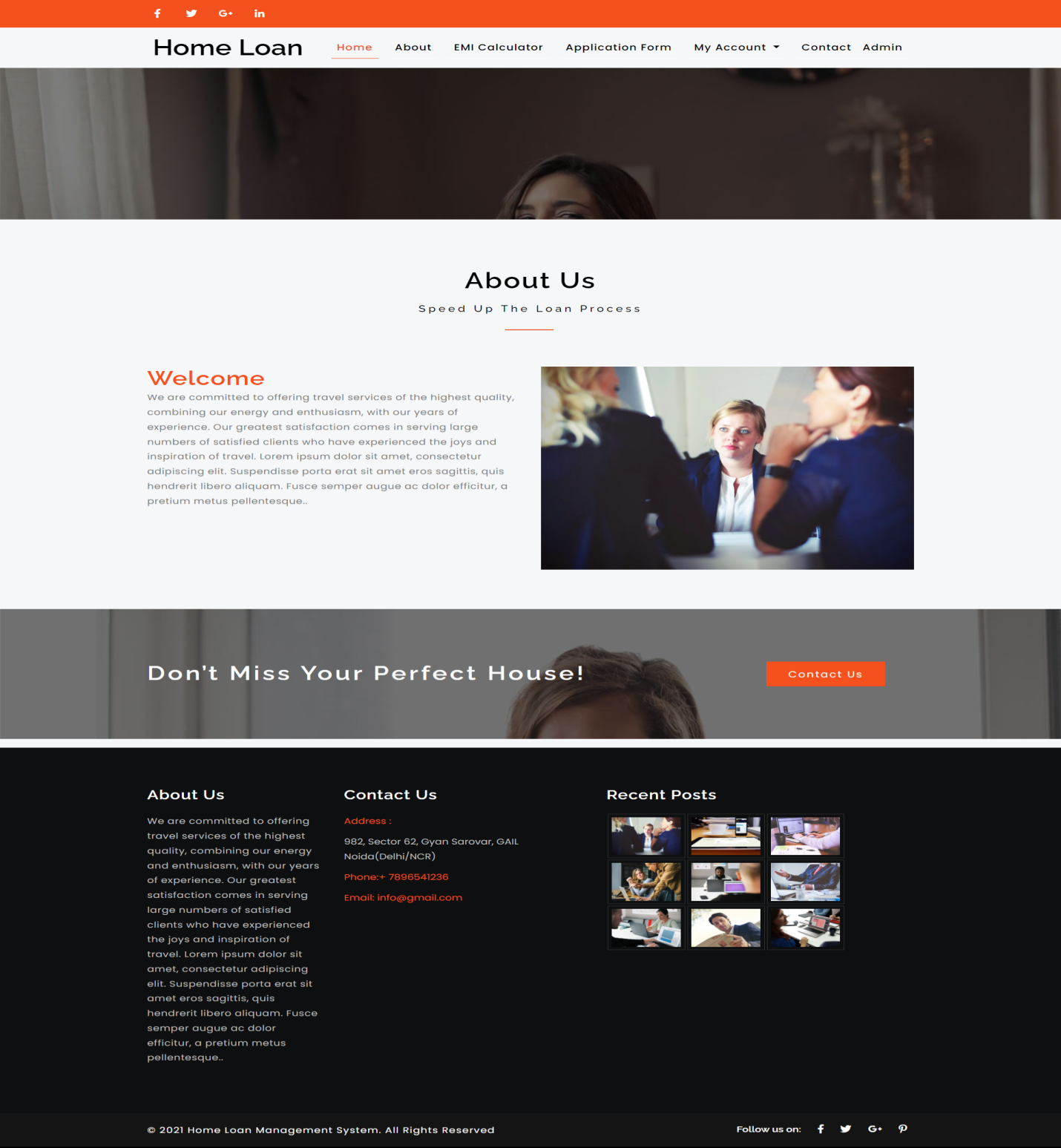
**CHAPTER-7**

**Screen Layout**

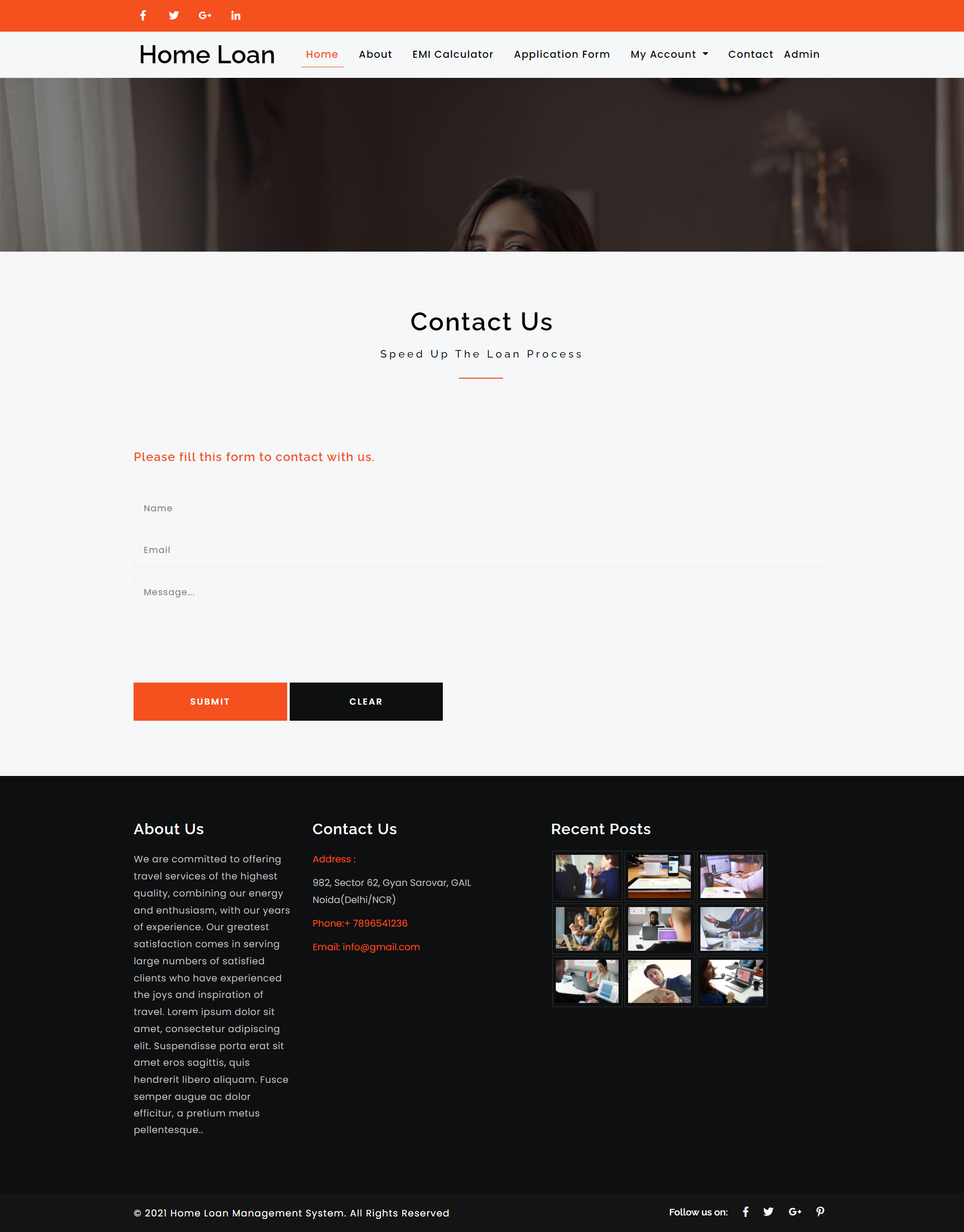
* **MAIN PAGE:**



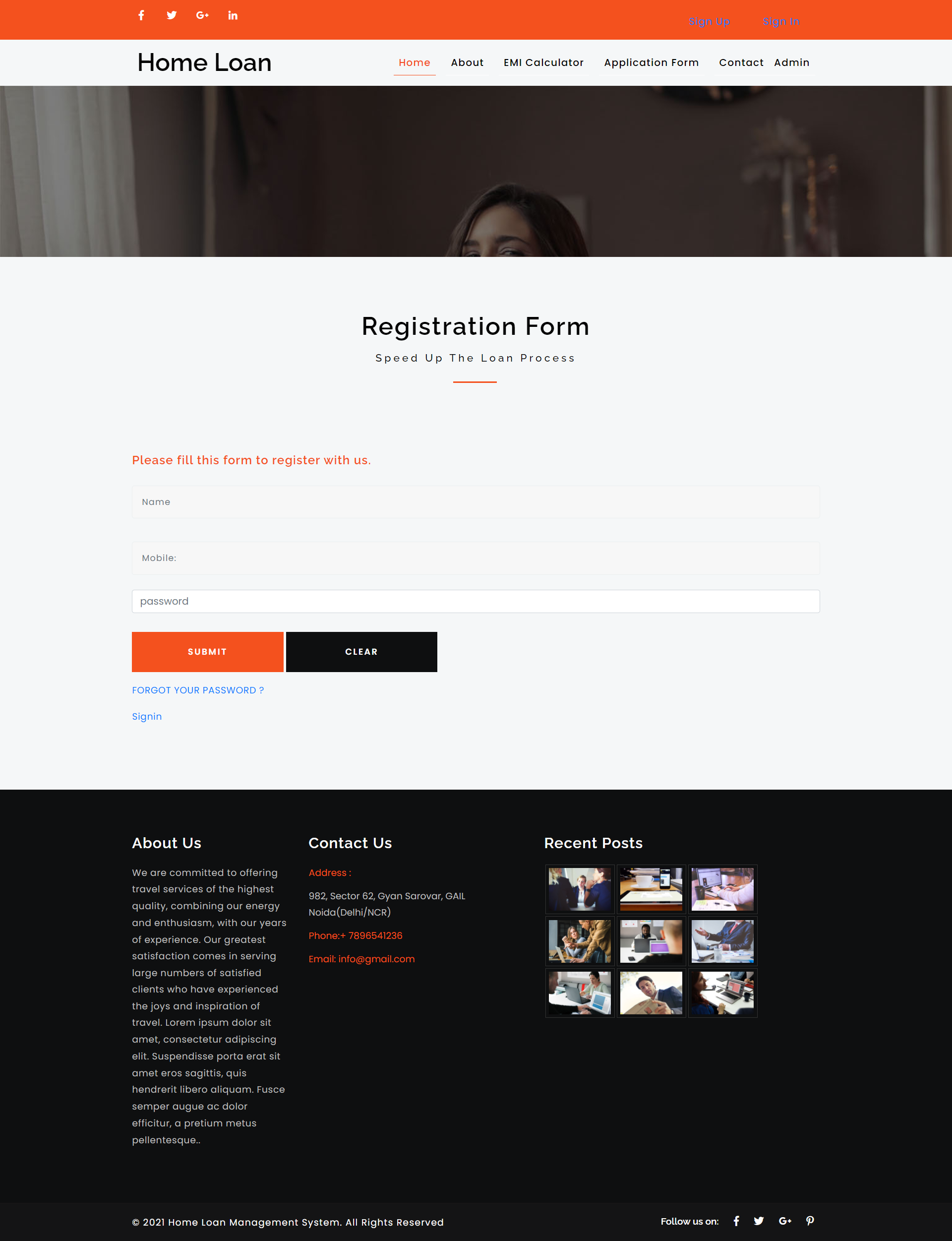
* **ADMIN PAGE:**



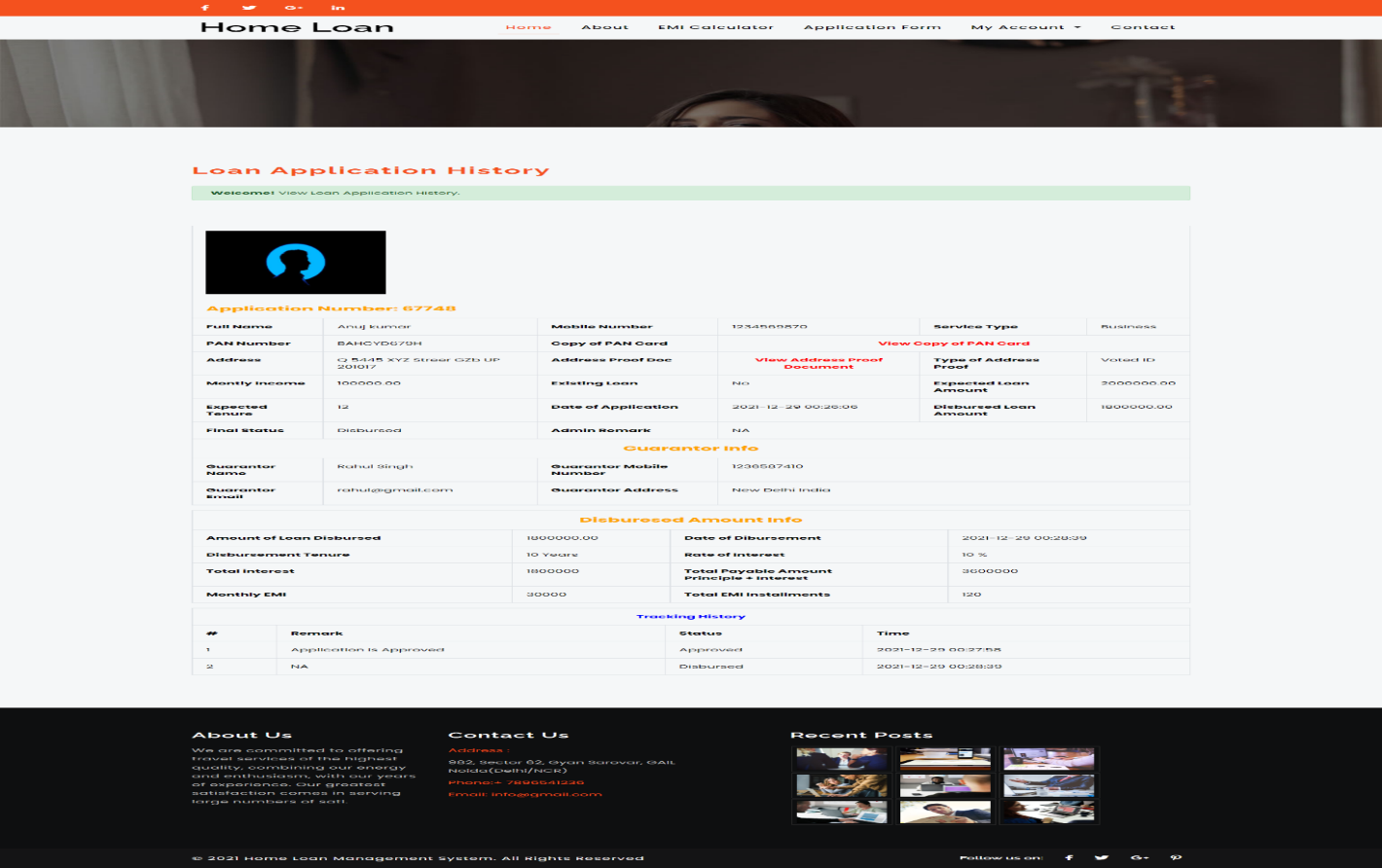
* **SIGN-UP PAGE:**



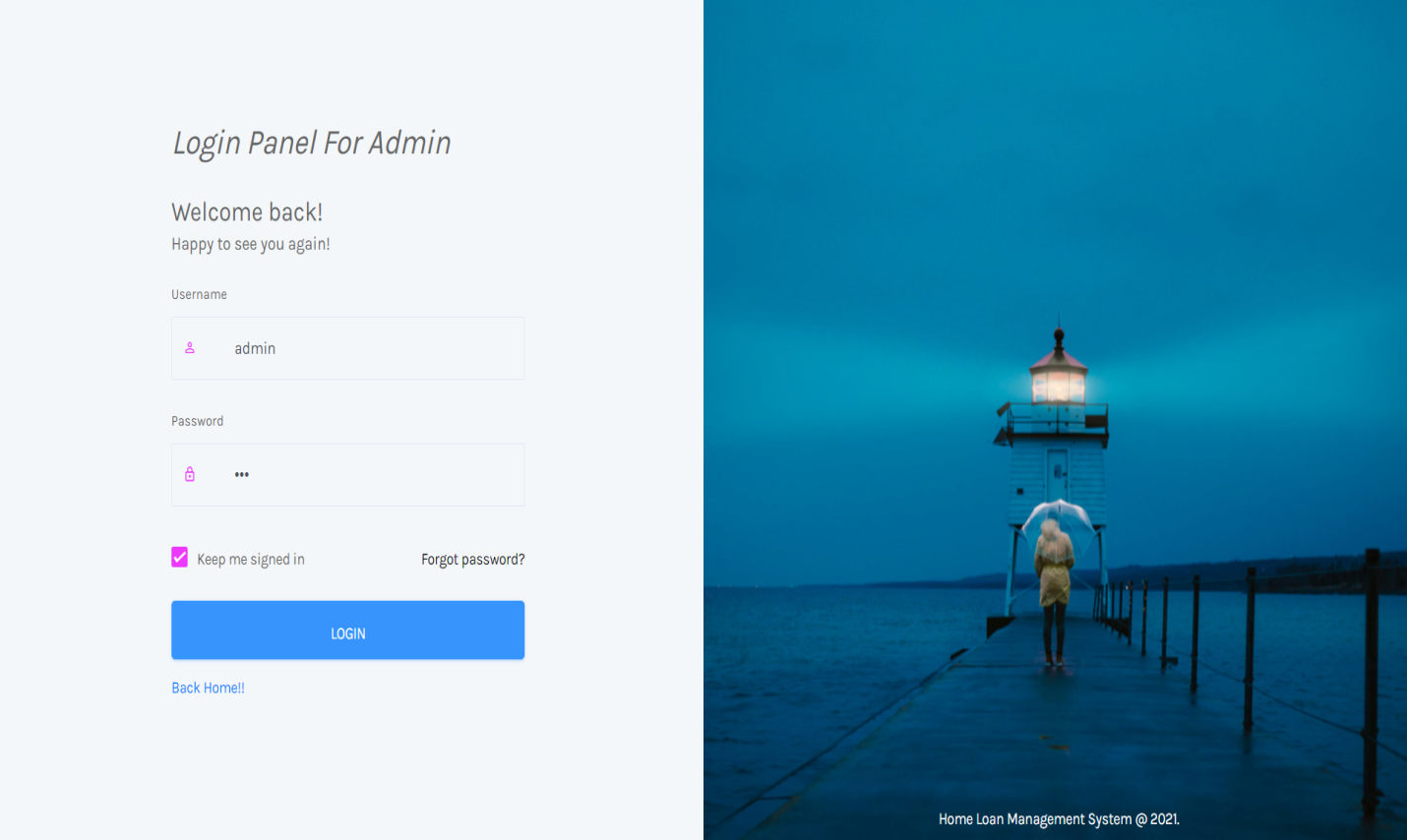
* **REGISTRATION PAGE:**



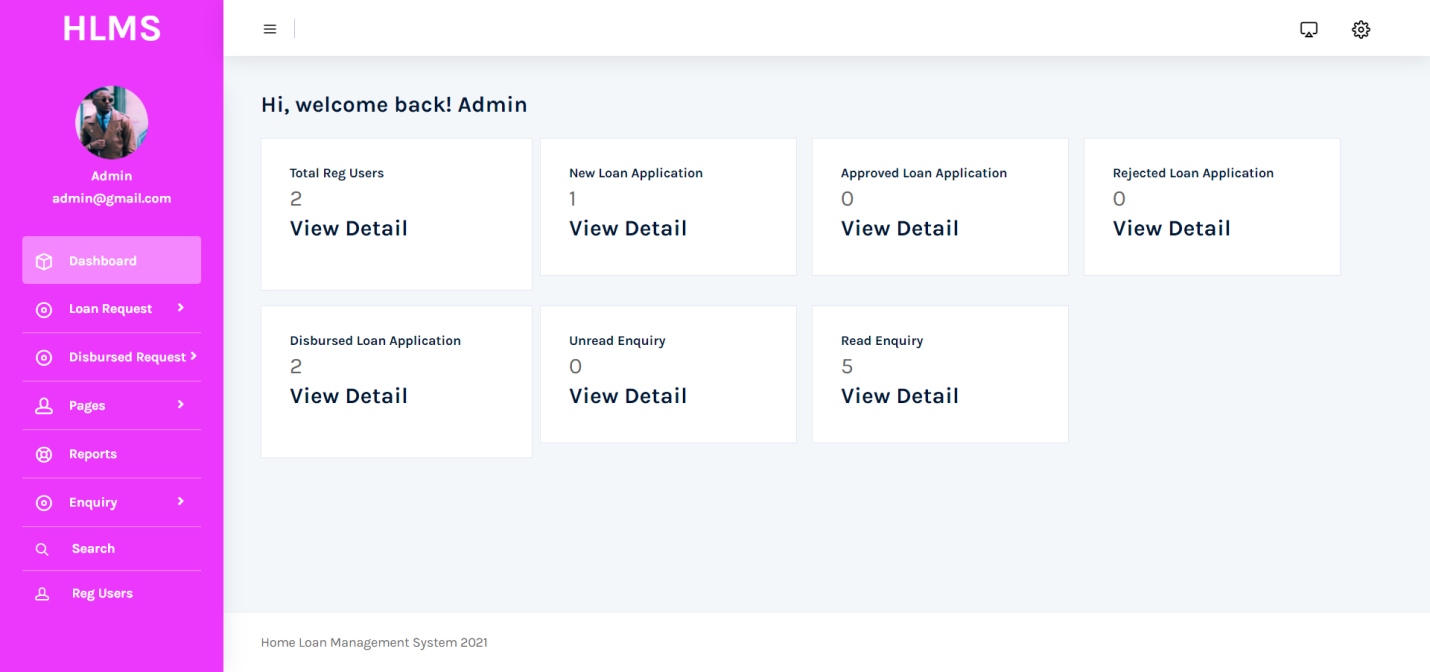
* **LOAN APPLICATION PAGE:**



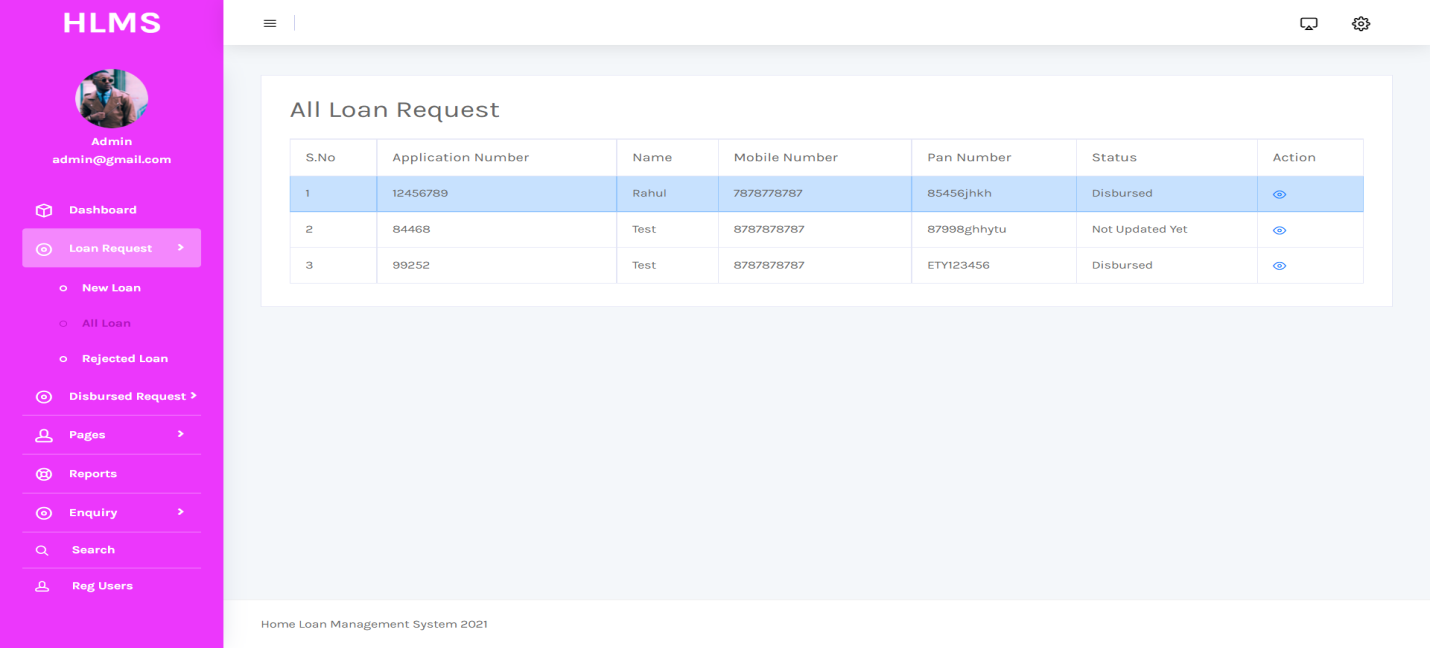
* **LOGIN PAGE:**



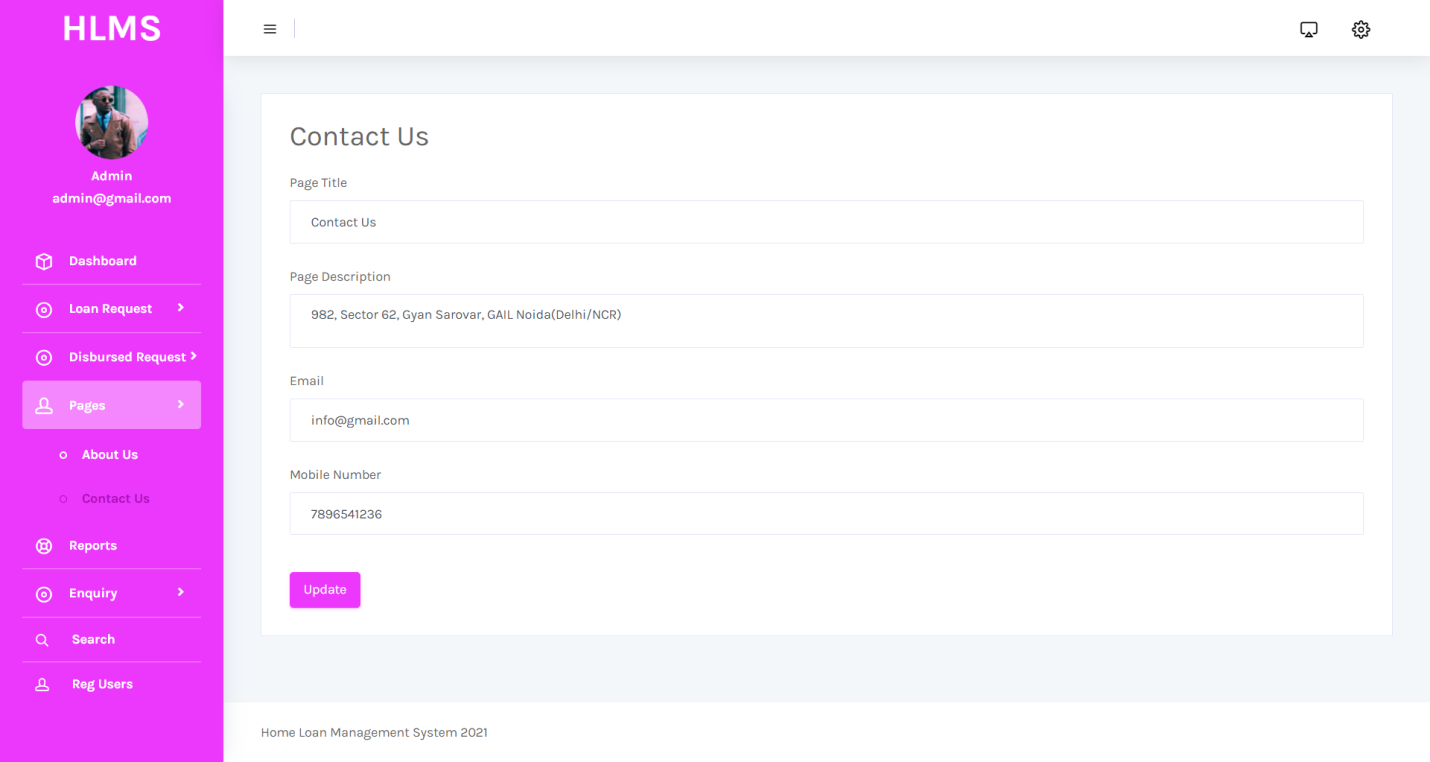
* **DASHBOARDPAGE:**



* **REQUEST PAGE:**



* **HELP PAGE:**



**CONCLUSION**

**CHAPTER-8**

**CONCLUSION**

* **Conclusion:**

The project titled as “Home Loan Website” was deeply studied and analyzed to design the code and implement. It was done under the guidance of the experienced project guide. All the current requirements and possibilities have been taken care during the project time.

Some existing system does not have facility to handle this loan system in a smooth manner. There is lot more to be done in order to provide a 100% secure platform for the loans which want to be taken with more description and also to make it more flexible.

To Conclude the Description of the Project. The Project is Developed using PHP,HTML,JS,CSS as Front-End and SQL Server as the Back-end and is based on the requirement specification of the user and the analysis of the Existing system with Flexibility.

**The Goals that are achieved by the Software are:**

* Optimum Utilization of Resources.
* Efficient Management of Records.
* Simplification of the records.
* Less Processing time and getting required information.
* User-Friendly.
* Portable and Flexible For further enhancement.

**FUTURE ENHANCEMENT**

**CHAPTER-9**

**FUTURE ENHANCEMENT**

* **Future Scope:**

Previously ,Data used to be inserted manually to analyse the result. But currently the project supports SQL files for extraction of data.The future scope is that data can be fetched parsed in another format like Doc, csv etc

This indicates that online purchasing and managing system is so convenient instead of manual work. We can do it in a better way by providing our software and the customer can purchase the products . This also provides high level of convenience for User and Admin.

**APPENDIX**

**CHAPTER-10**

**APPENDIX**

* SQL: Structured Query Language.
* HTML: Hyper Text Markup Language.
* PHP:Hypertext Pre-processor.
* JS:Java Script.
* CSS: Cascading Style Sheet
* DFD: Data Flow Diagram.
* ER: Entity Relationship.
* IDE: Integrated Development Environment.
* SRS: System Requirements Specifications.
* DOC: Document

**BIBLIOGRAPHY**

**CHAPTER-11**

**BIBLIOGRAPHY**

* **LINKS:**
* **https://www.youtube.com/watch?v=Sbskpj3OzWA**
* [**https://www.youtube.com/watch?v=cwDqjmSmtMQ**](https://www.youtube.com/watch?v=cwDqjmSmtMQ)
* [**https://www.php.net**](https://www.php.net)
* [**https://www.tutorialspoint.com**](https://www.tutorialspoint.com)
* [**https://www.javatpoint.com**](https://www.tutorialspoint.com)
* [**https://youtu.be/G3e-cpL7ofc**](https://youtu.be/G3e-cpL7ofc)
* [**https://youtu.be/1PnVor36\_40**](https://youtu.be/1PnVor36_40)
* **BOOKS:**
* A Complete overview on WEB-DEVELOPMENT by AYUSH MAURYAVANSHI
* Mastering HTML,CSS & JavaScript by Laura Lenay, Rafe Colburn.
* Static and Dynamic Webpage Development with HTML,CSS,PHP,JS,MYSQL by Bhumika S Zalavadia.
* Murach’s PHP and MySql (3rd Edition)
* Eloquent JavaScript 3rd Edition.