Loanalytic- Web

Software Requirements Specification

Version- 1.2

Date- 6/30/2019

Team Members

Gangadhar Adusumalli

Vamsee Krishna Gangapatnam

Keerthi Sree Kukunoor

Gouthami Pasham

Taraka Raviteja Peddi

Himabindu Poshala

Meghana Putta

Submitted in partial fulfillment

Of the requirements of

CSIS 44-691 Graduate Directed Project 1

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 6/11/2019 | Version 1 | Gouthami Pasham | First Revision |
| 6/13/2019 | Version 1.1 | Himabindu Poshala | Second Revision |
| 6/30/2019 | Version 1.2 | Meghana Putta | Third Revision |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Revision History 2](#_Toc12812767)

[Document Approval 2](#_Toc12812768)

[1. Introduction 5](#_Toc12812769)

[1.1 Purpose 5](#_Toc12812770)

[1.2 Scope 5](#_Toc12812771)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc12812772)

[1.4 References: 5](#_Toc12812773)

[1.5 Overview: 6](#_Toc12812774)

[2. General Description 6](#_Toc12812775)

[2.1 Product Perspective: 6](#_Toc12812776)

[2.2 Product Functions: 6](#_Toc12812777)

[2.3 User Characteristics 7](#_Toc12812778)

[2.4 General Constraints 7](#_Toc12812779)

[2.5 Assumptions and Dependencies 7](#_Toc12812780)

[3. Specific Requirements 7](#_Toc12812781)

[3.1 External Interface Requirements 7](#_Toc12812782)

[3.2 Functional Requirements: 8](#_Toc12812783)

[3.3 Use Cases: 9](#_Toc12812784)

[3.4 Class/Objects: 9](#_Toc12812785)

[3.5 Non-Functional Requirements 9](#_Toc12812786)

[3.5.1 Performance: 9](#_Toc12812787)

[3.5.2 Reliability 9](#_Toc12812788)

[3.5.3 Availability: 9](#_Toc12812789)

[Hosting the application on cloud to ensure it’s available to the user all the time. 9](#_Toc12812790)

[3.5.4 Security: 9](#_Toc12812791)

[3.5.5 Portability: 9](#_Toc12812792)

[3.6 Inverse Requirements 10](#_Toc12812793)

[3.7 Design Constraints 10](#_Toc12812794)

[3.8 Logical Database Requirements 10](#_Toc12812795)

[3.9 Other Requirements 10](#_Toc12812796)

[3.10 Prototypes (for complete project) 10](#_Toc12812797)

[3.11 Use Case Diagrams 17](#_Toc12812798)

[4. Design 18](#_Toc12812799)

[4.1 ER diagram: 18](#_Toc12812800)

[4.2 GUI 19](#_Toc12812801)

[5. Analysis Models 19](#_Toc12812802)

[5.1 Data Flow Diagram 19](#_Toc12812803)

[5.2 Sequence Diagram 19](#_Toc12812804)

[6. Technical Manual: 20](#_Toc12812805)

[6.1. Document Identification: 20](#_Toc12812806)

[6.2. System overview: 20](#_Toc12812807)

[6.3. Tools and technologies 20](#_Toc12812808)

[6.3.1. MySQL 20](#_Toc12812809)

[6.3.2 Express 20](#_Toc12812810)

[6.3.3 Plotly 20](#_Toc12812811)

[6.3.4 GitHub 20](#_Toc12812812)

[6.3.5 Heroku/AWS 20](#_Toc12812813)

[6.4 Document overview 21](#_Toc12812814)

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to build an online application- Loanalytic-Web for bank loans.

## 1.2 Scope

The purpose of Loanalytic-Web is to make the life of the borrower easy by simply registering to an online portal and get a loan approval. The application deals with 3 kinds of loans: Mortgage, Personal and Education loan. The user can quickly choose the kind of loan he wants and the amount he wants to borrow. Loan request would be validated by the application automatically as soon as the user answers few questions based on the kind of loan. The feasibility of loan approval would be displayed immediately. If the criteria is not met the user would be notified immediately else a background check is performed by the admin and a manual notification is sent by the admin. The application gives an opportunity to the admin to verify the user by validating the documents he has uploaded and doing some criminal/background check. Above all, we hope to provide a comfortable user experience along with the best pricing available.

## 1.3 Definitions, Acronyms, and Abbreviations

The following are the conventions used in the document:

* DB- Database
* ER- Entity Relationship
* IR- Interest Rate

## 1.4 References:

<https://www.quickenloans.com/l/progpi?gclid=Cj0KCQjwov3nBRDFARIsANgsdoEDTTowesJOqC8EQObf-jTuhdo9wy0lBS3gTbXcclNhE1gf7NtrIB8aArynEALw_wcB&qls=GAW_LNPREAPe.0000714218&ef_id=Cj0KCQjwov3nBRDFARIsANgsdoEDTTowesJOqC8EQObf-jTuhdo9wy0lBS3gTbXcclNhE1gf7NtrIB8aArynEALw_wcB:G:s&s_kwcid=AL!1083!3!43582782849!e!!g!!loan%20approval&gclsrc=aw.ds>

## 1.5 Overview:

# 2. General Description

## 2.1 Product Perspective:

This application enables the user to work with the bank virtually to find all the information to get the loan and gets his loan approved even without visiting the bank. The user need not travel to the bank to get his loan. The user can choose among various options available to him using the calculator available in the application.

## 2.2 Product Functions:

The application has two modes, the user mode and the Admin mode.

**User can perform all the below operations:**

* Register and login into the application.
* Change the password.
* Choose the kind of loan from the dropdown.
* Choose the amount he wants to borrow.
* Answer all the questions related to Eligibility criteria and background/criminal history verification.
* Choose the Credit score.
* Able to apply by submitting the required information and documents.
* Chat with customer support in case of any questions.
* Access the frequently asked questions document.
* Use the Loan Calculator to find the interest amount for various time periods and amounts.
* Ability to choose the Payment methods.
* Give a rating to the organization for their service.
* Able to view the status of their applications.

**Admin Can perform the below Operations:**

* Able to login to the website
* Able to add different categories of loans with requirements.
* View all the applications.
* View documents provided by the users.
* Manage users.
* Approve and reject loans
* Able to process loan and update status of loan applications.
* Count of application.
* Generate reports to view the customer payment history and more.

## 2.3 User Characteristics

This application is used by any users who wish to borrow amount for his immediate use.

## 2.4 General Constraints

Cannot think of any as of now.

## 2.5 Assumptions and Dependencies

All the developers have good understanding of the technology.

The project should be completed end of December, 2019.

# 3. Specific Requirements

## 3.1 External Interface Requirements

**3.1.1 User Interfaces**

The UI is developed using angular.

**3.1.2 Hardware Interfaces**

**3.1.3 Software Interfaces**

* Angular to develop the GUI
* MySQL database to store the data
* Plotly for generating reports.
* Heroku to host the application online.

**3.1.4 Communications Interface**

We can use email and chat box as a communicating interface.

## 3.2 Functional Requirements:

**Home page:**

* 1. This page contains a navigation bar having “chat” symbol, “sign in” and contact information.
  2. A simple dropdown with 3 options is available in the body of the home page.
  3. The 3 options in the dropdown are:
* Mortgage Loan
* Personal Loan
* Education Loan
  1. A Next button is available to navigate to the next page. Here a validation check should be performed. The user should not be allowed to navigate to the next page unless he chooses one of the option as listed above.
  2. User credit rating score should be displayed in the dropdown, allowing the user to choose a credit range. A next button and a back button is also available in same page allowing the user to navigate to the next page and previous page respectively. The user cannot navigate to the next page unless he choose his credit rating. User can navigate to the previous page irrespective of the option chosen or not chosen from the drop down.

These are the credit rating available in the dropdown:

* Low
* Below Average
* Average
* Above Average
* Good Excellent
  1. User is asked questions based on different loans and is allowed to choose the period of repayment.
  2. Based on the repayment period, the total amount to be payed, the interval and the complete summary of the loan is displayed on the page along with an Accept button, back button and cancel button.
  3. If the cancel button is clicked the application should display the initial page.
  4. If the back button is clicked, navigate to the previous page.
  5. The user can click on accept button, if he likes the quote. If the users clicks on Accept he is asked few questions to register or login into the application. Register or login into the application to upload few documents. 2 attachments to upload the documents, next and back button should be provided.

Here is the list of document the user is asked to upload:

* Previous 3 months pay slips
* Id proofs
  1. Clicking on the next should give a notification to the user, conforming the application is filled and an email should be sent to the user.
  2. There is a chat button available in the navigation bar. It has to open the chat box and allow the user to talk to the admin in case of any questions.
  3. There is a sign in button available on the navigation bar. Clicking on the sign in button allows the user or admin to log in into the application.
  4. If the user logs in, the status of his application and the previous application history should be displayed. This page should also have a button which should redirect to the step 2.
  5. If the admin logs in, he can view various options in the body of the page:
* View applications.
* Create new loans.
* View the trend of the payments.
  1. The Admin can view each application, access all the documents uploaded by the user and approve/reject the application accordingly.
  2. The admin has the ability to send an email to the user from the application.

**Signup/Sign in Page:** This allows the user or admin to log in into the application.

**Forgot Password:** This allows the user to change the password. The link to change the password should be sent to his email id provided during registration.

## 3.3 Use Cases:

## 3.4 Class/Objects:

## 3.5 Non-Functional Requirements

### 3.5.1 Performance:

Using data optimization techniques to retrieve the data quickly.

### 3.5.2 Reliability

Ensure to use ACID rules while designing the database, thereby making reliable system.

### 3.5.3 Availability:

### Hosting the application on cloud to ensure it’s available to the user all the time.

### 3.5.4 Security:

The info provided by the user is secure and would not be shared to anyone.

### 3.5.5 Portability:

Application is designed in such a way that it can be accessed by any kid of devices

## 3.6 Inverse Requirements

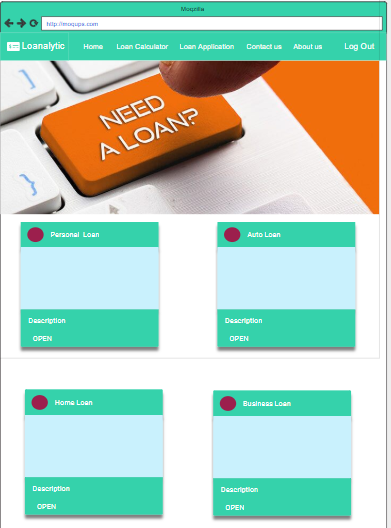
## 3.7 Design Constraints

## 3.8 Logical Database Requirements

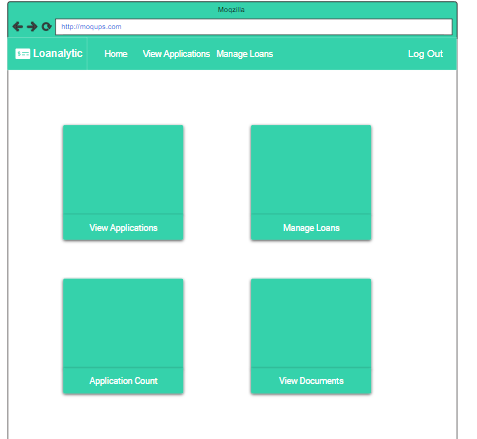
## 3.9 Other Requirements

## 3.10 Prototypes (for complete project)

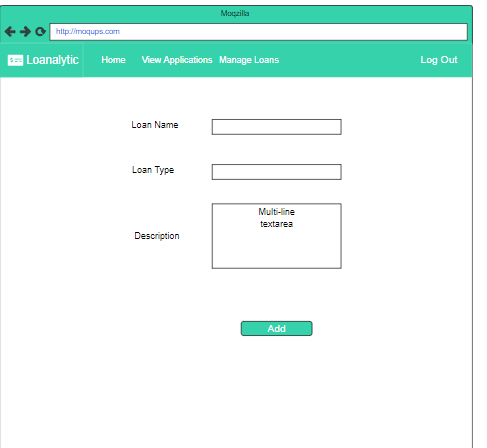
Home Page:-

****

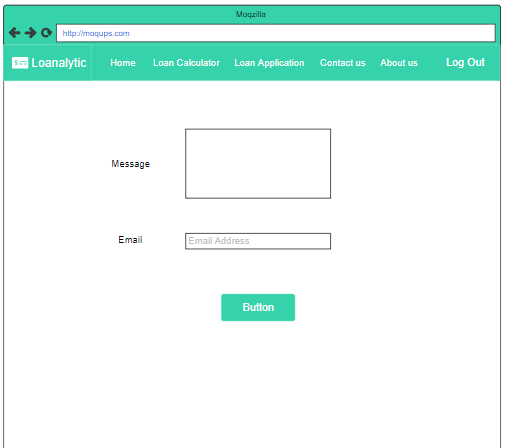
Admin Home:-

****

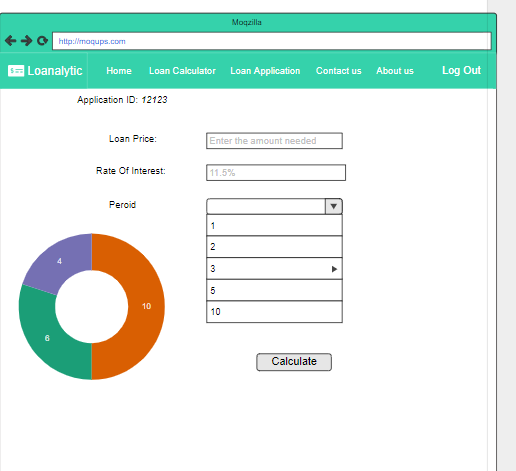
Manage Loans:-



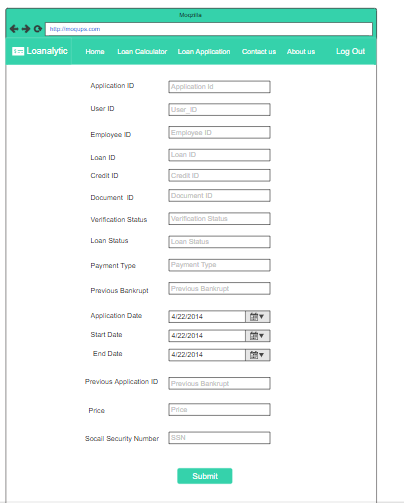
Contact Us :-



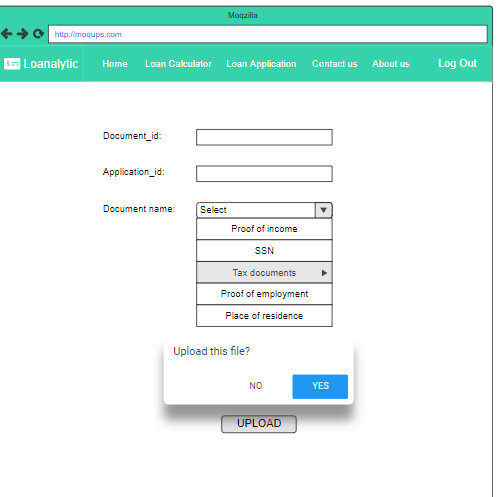
Loans:-



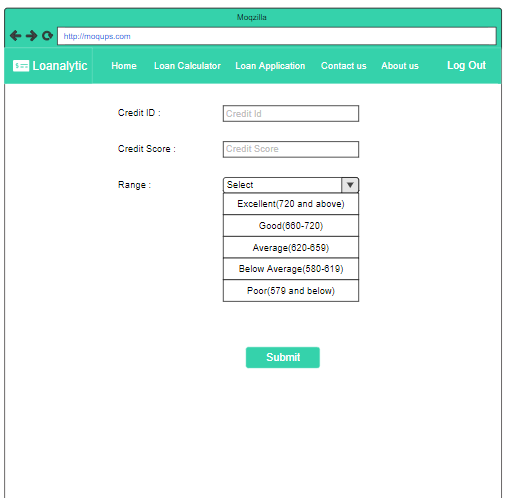
Loan Application:-



Loan Documents:-



Credit Score:-



## 3.11 Use Case Diagrams

# 4. Design

## 4.1 ER diagram:

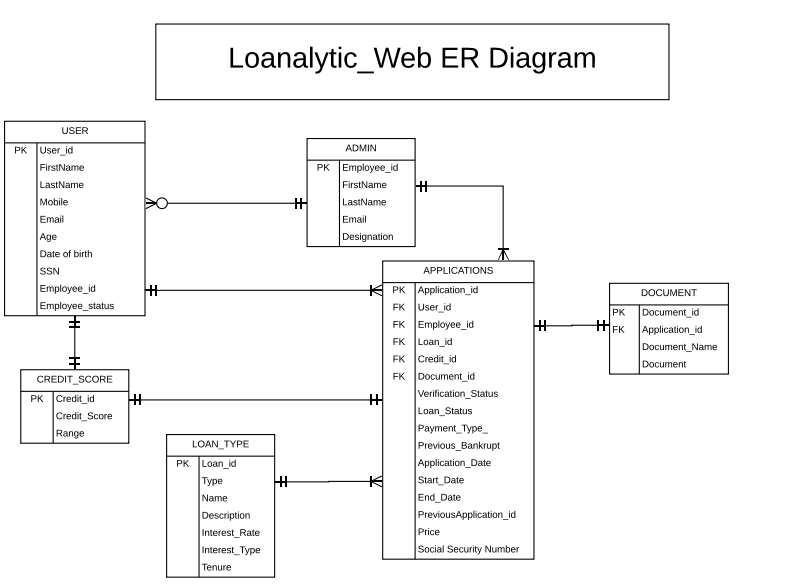
The major features of Loanalytic database system as shown in the below diagram.

Fig 4.1

Fig 4.1 gives a complete description of all the entities used in the application. The description of each entity and the relation between them is specified below:

**USER:**

The user entity has all the attributes to store the information related to the user. It stores the information provided by the user during the registration phase.

**ADMIN:**

The admin entity stores the details related to the employees working for the application. The attributes include the employee id, first name, last name, email and designation.

**DOCUMENT:**

Document entity stores the required documents uploaded by the user and later the admin can view these documents.

**APPLICATIONS:**

Application table has all the information related to the application.

**LOAN TYPE:**

Loan type is a master table, it has all the in information related to loans. Admin can introduce new loans into the system through the interface.

**CREDIT SCORE:**

Credit score is a master entity, it stores the basic information of Credit score and the range associated with it.

## 4.2 GUI

# 5. Analysis Models

## 5.1 Data Flow Diagram

## 5.2 Sequence Diagram

# 6. Technical Manual

## 6.1. Document Identification

This section of the document describes the technical details of the project “Loanalytic”. The tools/technologies used are listed. The code is explained in detail which helps any developer to take over the project with minimal knowledge transfer or trainings.

## 6.2. System overview

Loanalytic helps any user to apply for the loan. As of now the application is used for 3 different kind of loans.

1. Education Loan
2. Personal Loan
3. Mortgage Loan

The application is designed in such a way that the developer can add additional loan types based on the requirements. The functional logic and calculation of 3 loan types is described in this document. Anything related to future loan types is not the scope of this document.

## 6.3. Tools and technologies

### 6.3.1. MySQL

MySQL is used as a database. Various optimization techniques index creation on tables are used to retrieve the data quickly.

### 6.3.2 Express

Express is a used as web application framework for Node. It handles the interactions between the frontend and the database.

### 6.3.3 Plotly

Banking system requires a lot of reporting to maintain the assets. Visualizing the data helps the loan provider monitor his assets with ease. Plotly is used to generate the reporting functionality for the admin.

### 6.3.4 GitHub

GitHub is used as a version control tool. All the code and document related to the project is uploaded into the GitHub repository. We have 8 contributors working on the project as of now.

GitHub link: <https://github.com/Gouthami-pasham/Loanalytic>

### 6.3.5 Heroku/AWS

Heroku is used to host the application online. The application might also be hosted on AWS based on the budget availability. (Decision yet to be made)

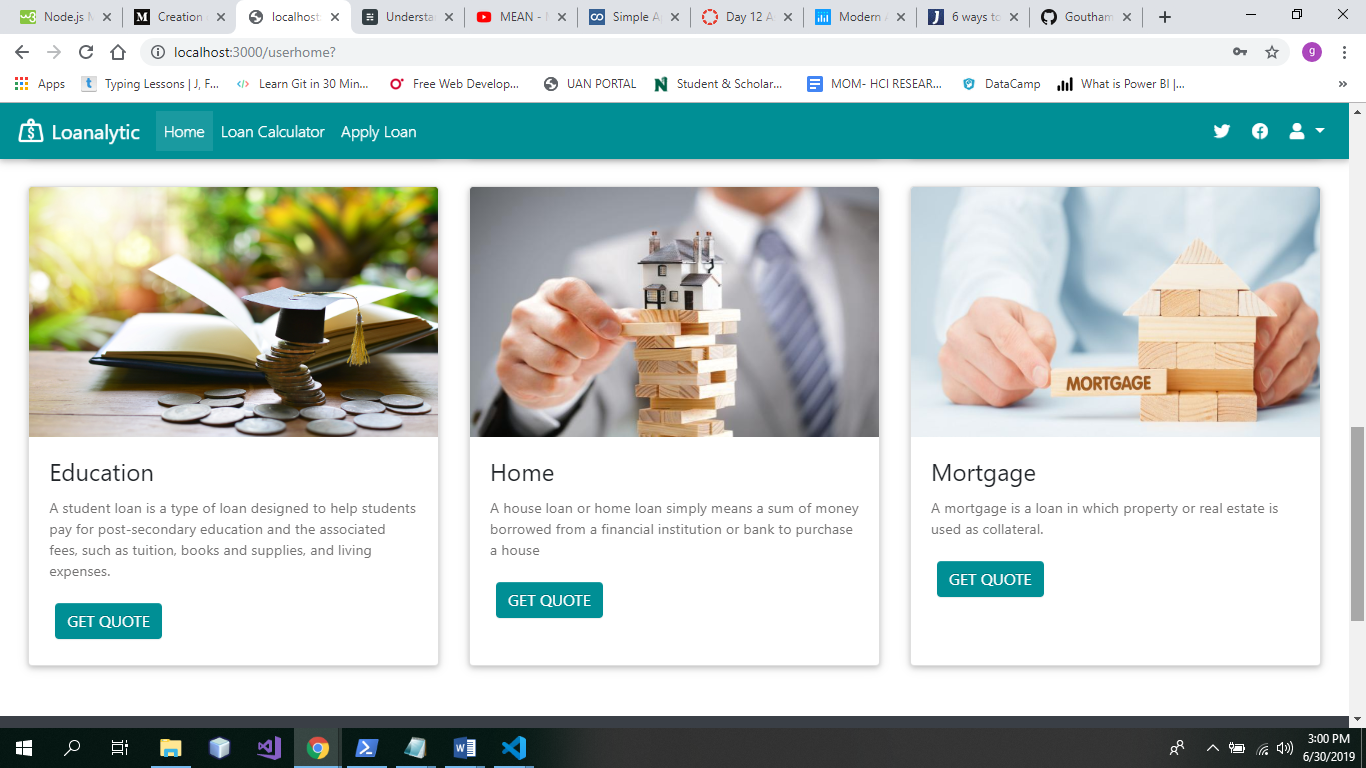
## 6.4 Document overview

This section of the document lists each component of the application along with detail explanation of code and the screenshot. The complete project is not implemented in this semester so the major objective of this document is to explain the login functionality.

* User login
* Admin login
* Registration page
* Forgot password
* User home page
* Admin home page
* Loan calculator
* Contact us
* Chat
* View applications
* Manage user Installments Payments

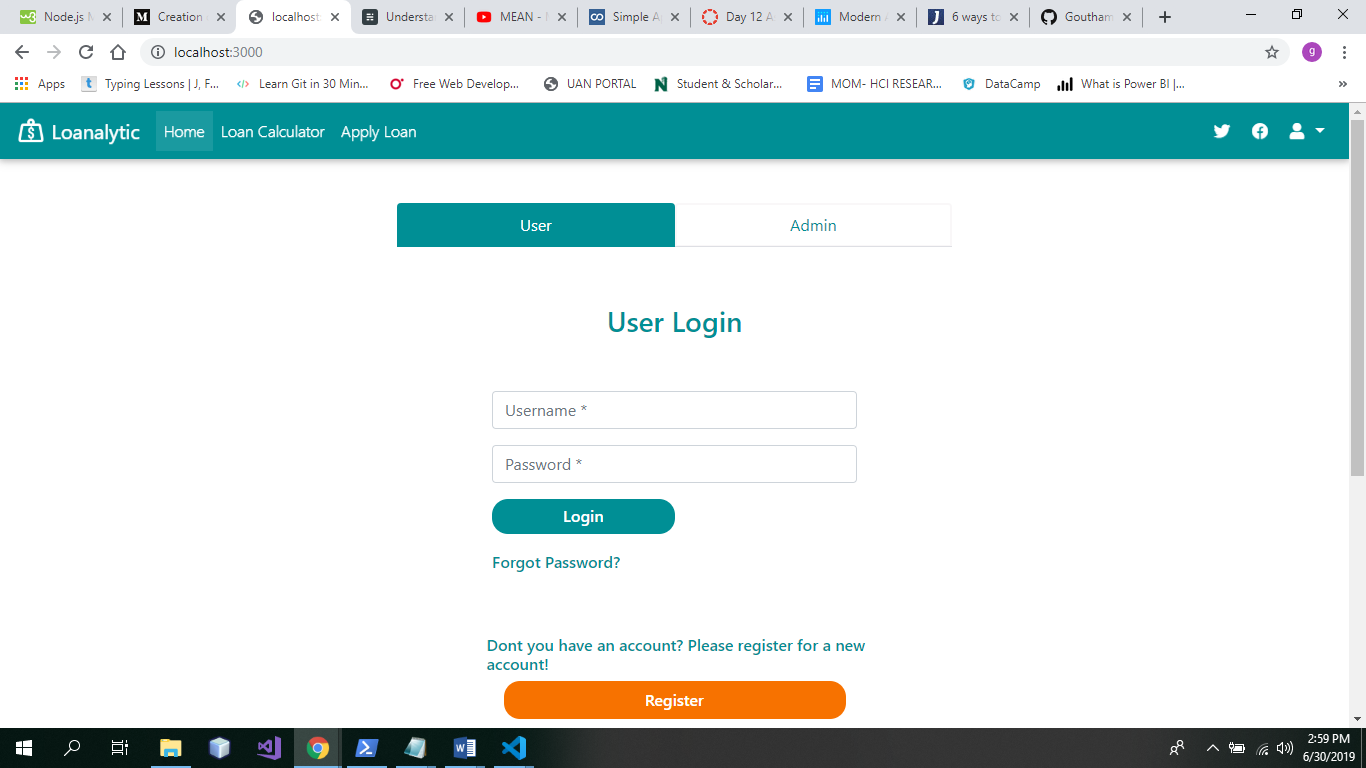
**Home Page:**

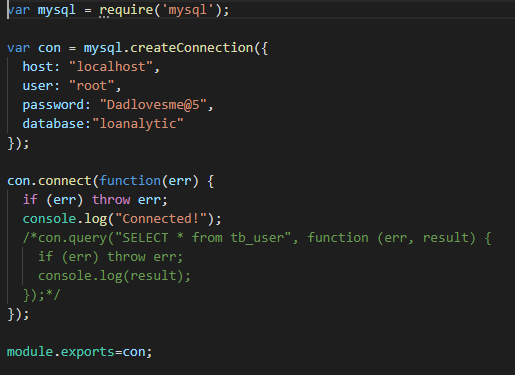
Any user trying to access the application can view the home page and select the kind of loan he would like to apply. By clicking on the available options, as shown in the below home page, the user is navigated to a different page to fill in the details required for loan processing. The functionality of this page is not built yet the design of the page looks as shown in the screenshot below. The HTML code for this page is available in the GitHub page: userhome.ejs under views folder.



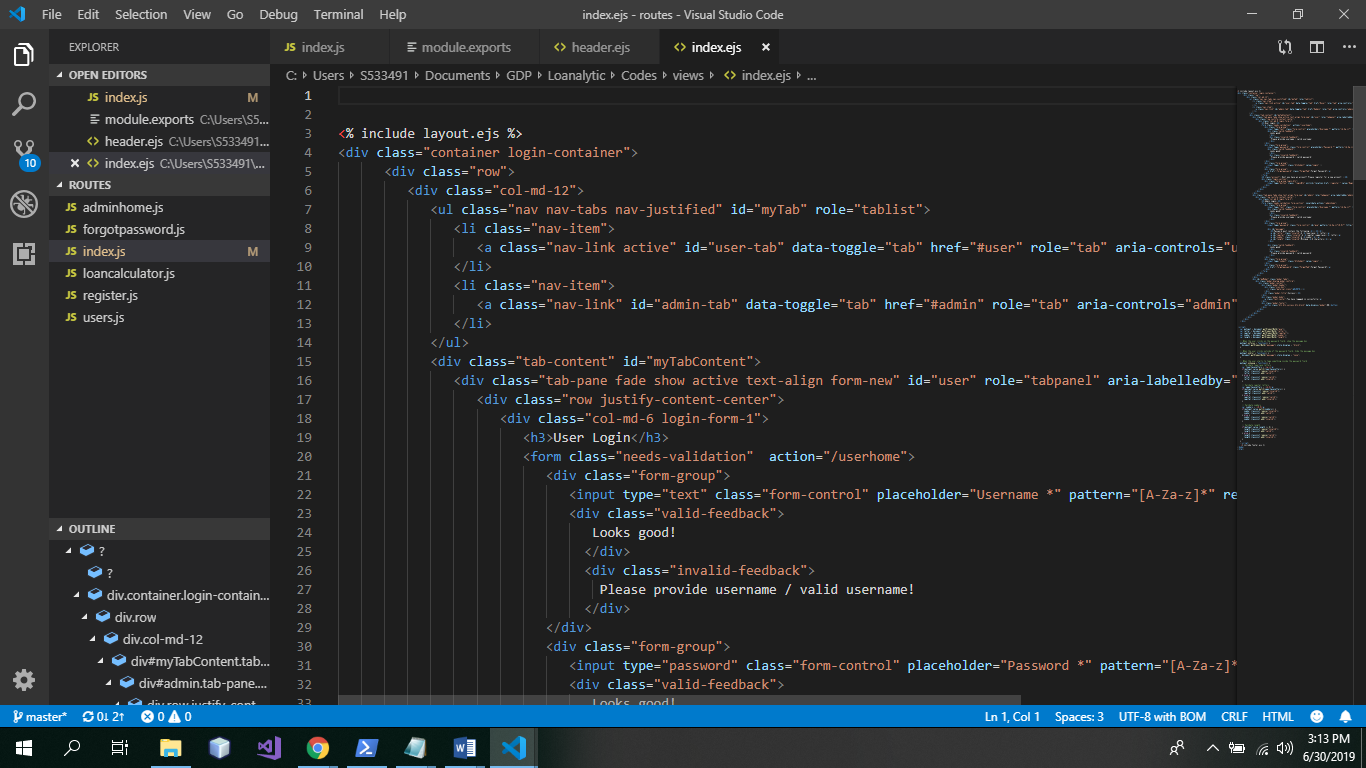
**User Login:**

If the customer is already registered with the application. He/she can login into the system to view the status of the application or view the payment information and make payments. The login details of the user is stored in the MySQL database. The code available in db.js is used to connect to the database. The database name, host user and password are stored in this file.

****

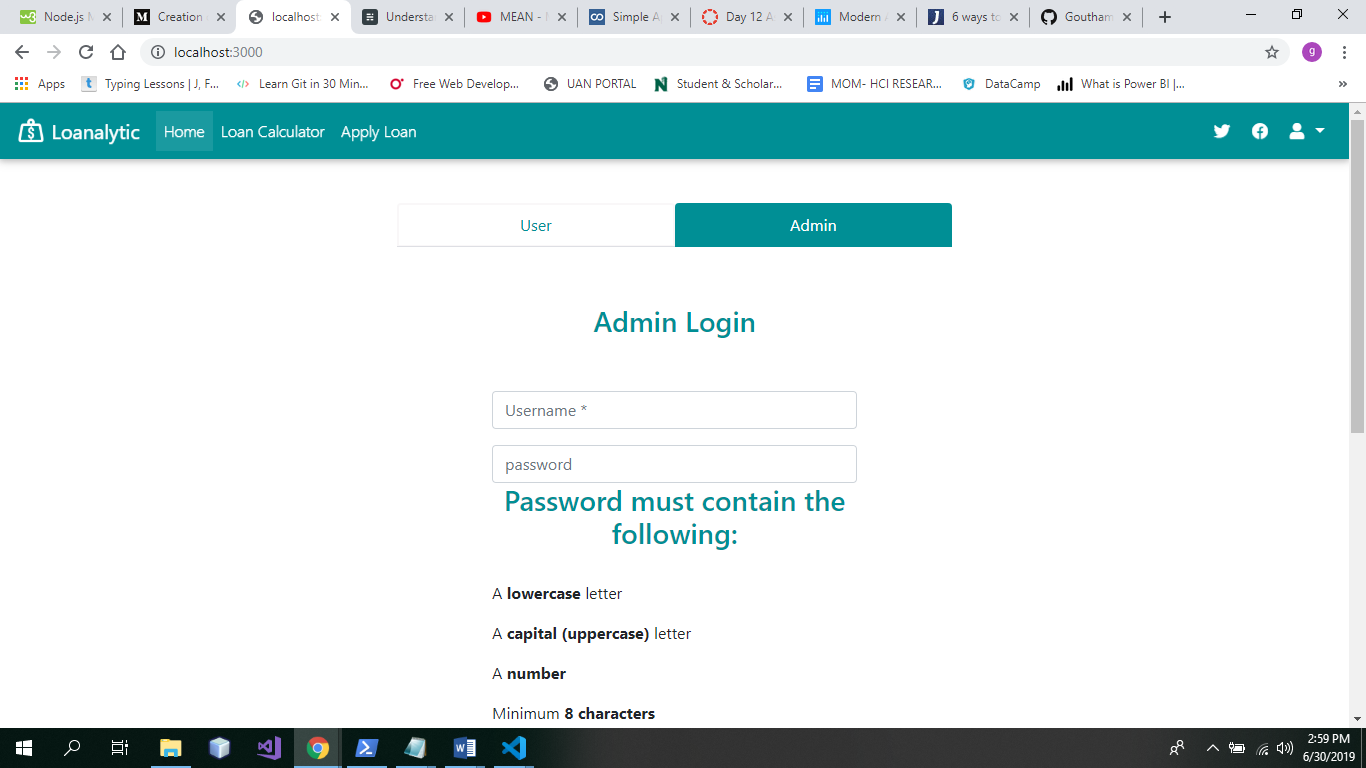


The application is build using the MVC architecture. The code snippet below shows the script for building the login page. Bootstrap is used to design the front end using various designs available to beatify the application.



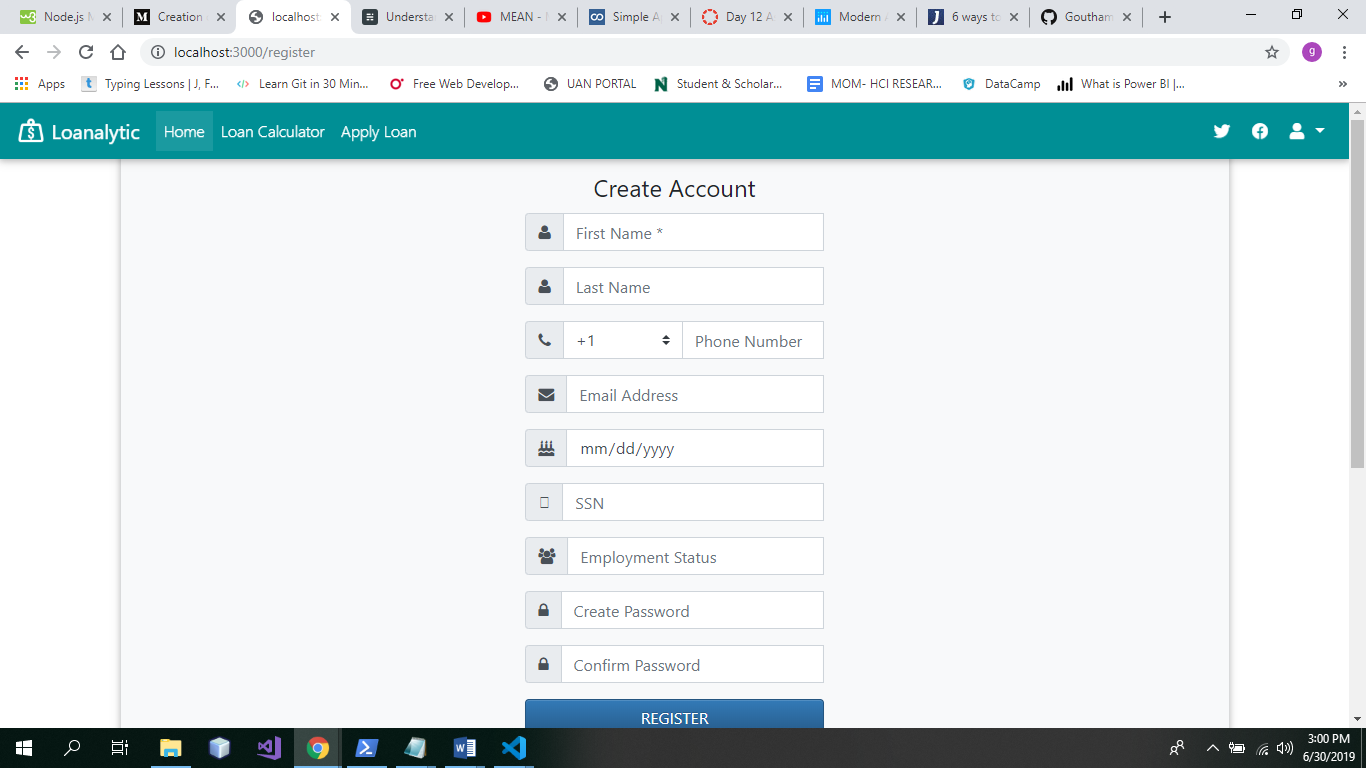
**Admin Login:**

The admin can use the Admin login page to login in to the application. As soon as the admin is logged into the application he/she can perform admin related functionalities as specified in the functional document.

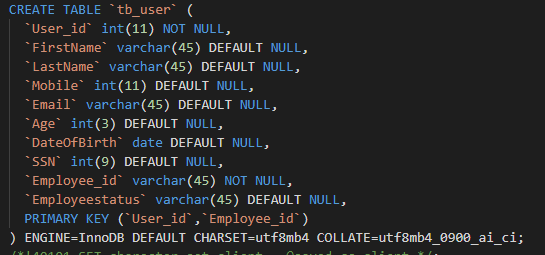


**Registration Page:**

New user can register using the registration page. The code is available in register.ejs file under views folder.



The user registration details are stored in the Tb\_user table in the database. The below screenshot shows the script for Tb\_user. To connect the front end to database we have created the model, controller. The model represents the shape of the data and is available in the register.js file in the model folder.



Check if the data is inserted into the database via the postman application. Connect to postman and give the following URL: <http://localhost:3000/register/createuser>

