Shri Sairaj Shikshan Pratisthan’s

Vishwalata Art’s, Commerce and Science

College, Bhatgaon, Tal. -Yeola, Dist.-Nashik.

**(Affiliated With SPPU)**

**Department Of Computer Science**



Project Report On :

***Credit Card Application Management System***

Represented By :

Mr.Chavan om mohan

Mr.Miskin omkar rajesh

# From : T.Y.B.Sc.(Computer Science)

Under The Guidance Of :

**Prof. Wagh T.B.**

**Shri Sairaj Shikshan Pratisthan’s**

**Vishwalata Art’s, Commerce**   **and Science**

**College, Bhatgaon, Tal.-Yeola, Dist.-Nashik.**

**(Affiliated With SPPU)**

**Department Of Computer Science**



# CERTIFICATE

**This is to certify that, MR. Chavan Om Mohan(Roll No.-6).**

**, MR. Miskin Omkar Rajesh(Roll No.- )Has completed the project report on*, Credit Card Application Management System***

**As per syllabus, told down by**

**University of Pune for B.Sc. (Computer Science) as year 2022-2023.**

**Prof. Wagh T.B.** **Prof. Wagh T.B.**

**(Project Guide) (Head of Department)**

**Internal External**

**DEPARTMENT OF COMPUTER SCIENCE**

**2022-23**

**“Credit Card Application Management System”**

**SUBMITTED BY**

**MR. OM M. CHAVAN**

**&**

**MR. OMKAR R. MISKIN**

**CLASS**

**TY BSC (COMPUTER SCIENCE)**

**GUIDED BY**

**PROF.**

**“SHRI SAIRAJ SHIKSHAN PRATISHTHAN’S”**

**“VISHWALATA ARTS, COMMERCE, SCIENCE COLLAGE “**

**BAHTGAON, TAL YEOLA(NSHIK)**

**Credit Card Application Management System**

**Developed**

**Using**

**PHP and MySQL**

**Project Report**

**By**

**ACKNOWLEDGEMENT**

I extend my deepest appreciation to my esteemed guide, Mr. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for providing me with the possibility to complete this project with the right guidance and advice.

Special gratitude I give to my respected head of the division Mr.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, for allowing me to use the facilities available and also help me to coordinate my project

Furthermore, I would also like to acknowledge with much appreciation the crucial role of faculty members on this occasion.

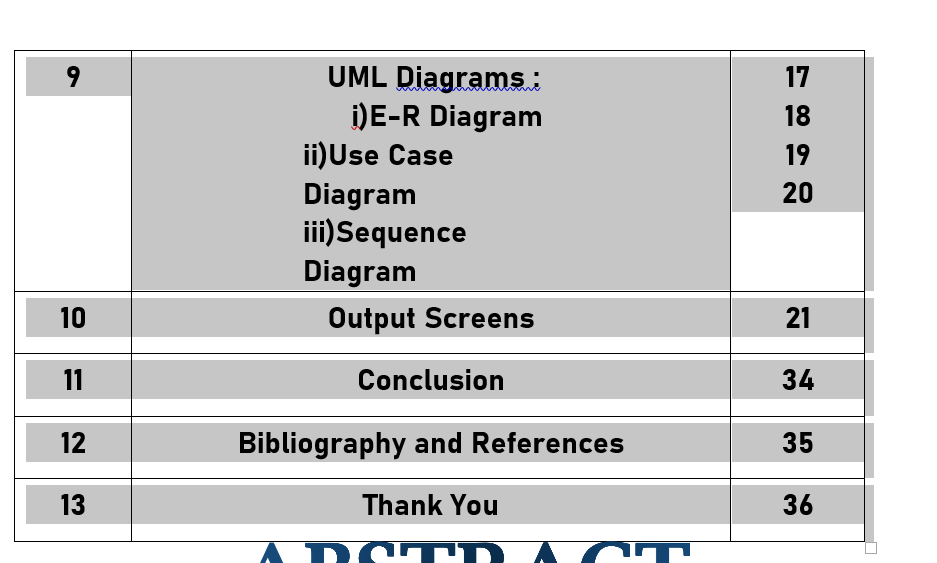
Last but not least, I would like to thank friends who help me to assemble the parts and gave a suggestion about the project.

.

**----------------------**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR.No.** | **TITLE** | **PAGE NO.** |
| **1** | **Abstract** |  |
| **2** | **Problem Statement** |  |
| **3** | **Purpose / Objective or Goals** |  |
| **4** | **Project Scope and Limitation** |  |
| **5** | **Brief Overview of The Technology** |  |
| **6** | **Feasibility Study** |  |
| **7** | **System Analysis :**  **i)Existing System ii)Proposed System** |  |
| **8** | **Database Design** |  |



**Abstract**

Credit Card Application Management System contains data and information of credit card holder who apply for credit card online. The main purpose of CCAMS is to systematically record, store and update the details of admin/sub-admin and also manage the credit card application.

This projects helps to those people who apply credit card online without wasting a time and also check the application status by their name/mobile number/application number.

**Introduction**

**Introduction:-**

Credit Card Application Management System is a web-based technology the main purpose this project is to provide all online credit card application management. User can apply credit card online and check the application status by using his/her name /mobile number/application number.

In Credit Card Application Management System we use PHP and MySQL database. This is the project which keeps records of of admin/subadmin and also manage the credit card application. Credit Card Application Management System has three module i.e. admin, Sub-Admins and users.

**Admin Module**

**Dashboard**: In this section admin can briefly view total number of Subadmins and total number of new application, accepted application and rejected application.

**Subadmins**: In this section, admin can manage the Sub-Admins (add/update,Delete).

**CC Application**: In this section, admin can manage the credit card application and change the status of application according to current scenario.

**Report**: In this section, two reports are available.

* **B/w Dates Report:** Admin can view number of credit card application received in the particular periods.
* **Search Report:** admin can search Credit Card Application by using Name/Email/Mobile Number/Application Number.

**Pages:** In this Section, Admin can manage the content of about us and contact us page.

**Account Settings:**

* **Profile**: In this section admin can update his/her profile.
* **Change Password**: In this section admin can change his/her own passwords
* **Logout**: Through this button admin can logout.

**Forgot Password:** In this section, admin can reset his/her password by using registered email id and contact number.

**Sub-Admin Module**

Sub-Admin and Admin features are the same except Sub-Admin creation. Sub-Admin can’t create the Sub-Admins.

**Note**: In this project MD5 encryption method used.

**Users**

User can visit the website.

User apply for credit card online.

Users can check the application status by using application number/name/mobile number.

**Purpose:-**

The purpose of developing Credit Card Application Management System is to apply credit card online without wasting a time. Another purpose for developing this application is to generate the report automatically.

**Scope:-**

Credit Card Application Management System is developed as a web application and it will work over web to apply credit card online.

**Requirement Specification**

**Hardware Configuration :**

**Client Side:**

|  |  |
| --- | --- |
| **RAM RAMfgdfRA RAM** | 512 MB |
|  |  |
| **Hard disk** | 10 GB |
|  |  |
| **Processor** | 1.0 GHz |
|  |  |

**Server side:**

|  |  |
| --- | --- |
| **RAM** | **1 GB** |
| **Hard disk** | **20 GB** |
| **Processor** | **2.0 GHz** |

**Software Requirement:**

**Client Side:**

|  |  |
| --- | --- |
| **Web Browser** | Google Chrome or any compatible browser |
| **Operating System** | Windows or any equivalent OS |

**Server Side:**

|  |  |
| --- | --- |
| **Web Server** | APACHE |
| **Server side Language** | PHP5.6 or above version |
| **Database Server** | MYSQL |
| **Web Browser** | Google Chrome or any compatible browser |
| **Operating System** | Windows or any equivalent OS |

**APACHE**

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 20th birthday as a project in February 2015.

**PHP**

* PHP stands for PHP: Hypertext Preprocessor.
* PHP is a server-side scripting language, like ASP.
* PHP scripts are executed on the server.
* PHP supports many databases (MYSQL, Informix, Oracle, Sybase, Solid, Generic ODBC, etc.).
* PHP is an open source software.
* PHP is free to download and use.

**MYSQL**

* MYSQL is a database server
* MYSQL is ideal for both small and large applications
* MYSQL supports standard SQL
* MYSQL compiles on a number of platforms
* MYSQL is free to download and use
* How to access MySQL: <http://localhost/phpmyadmin>

**Analysis and Design**

**Analysis:**

Customer can apply for credit card and know his eligibility from his own place just by giving his personal details. Everything comes to his door. Person comes to collect the required documents.

**Disadvantage of present system:**

* **Not user friendly:** The present system not user friendly because data is not stored in structure and proper format.
* **Manual Control:** All report calculation is done manually so there is a chance of error.
* **Lots of paper work:** Lawyers/Advocates record maintain in the register so lots of paper require storing details.
* **Time consuming**

**Design Introduction:**

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data

UML Diagrams:

Actor:  
 A coherent set of roles that users of use cases play when interacting with the use `cases.

Use case: A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

**USECASE DIAGRAMS:**

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what’s called an actor.

Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can’t do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

* The purpose is to show the interactions between the use case and actor.
* To represent the system requirements from user’s perspective.
* An actor could be the end-user of the system or an external system.

**USECASE DIAGRAM:** A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

**Use Case Diagrams:**

**Admin**

**Sub-Admin**

**Users**

**User**

**ER Diagram:**

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

* It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

**ER Notation**

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used; among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

* **Entities** are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
* **Relationships** are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs
* **Attributes**, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
* **Cardinality** of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

**Existence** is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

****

# Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

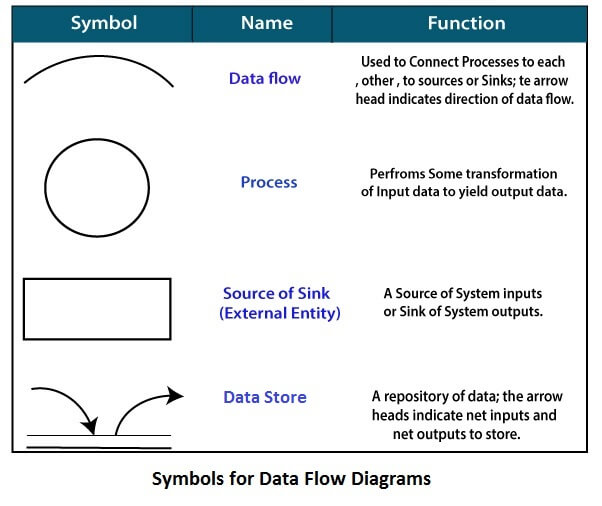
It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

**The following observations about DFDs are essential:**

1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.
4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

Standard symbols for DFDs are derived from the electric circuit diagram analysis and are shown in fig:

******

**Circle**: A circle (bubble) shows a process that transforms data inputs into data outputs.

**Data Flow**: A curved line shows the flow of data into or out of a process or data store.

**Data Store**: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.

**Source or Sink**: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

**Zero Level DFD**

**Login Management**

**Generate Report**

**Authorization Management**

**Admin**

**Management**

**Application Request Management**

**Sub-Admin**

**Management**

**Changing Password**

**Management**

**Password Management**

**Changing Password**

**Management**

**Application Management**

**Authorization Management**

**Sub-Admin Management**

**Generate Report**

**Login Management**

**Search Application**

**Password Management**

**Admin**

**Management**

**Application**

**Management**

**Second Level DFD**

**Admin**

**Manage Sub-Admin**

**Manage CC Application**

**Search Application**

**Generate B/w dates Report**

**Update Profile**

**Change Password**

**Sub-Admin**

**Manage CC Application**

**Search Application**

**Generate B/w dates Report**

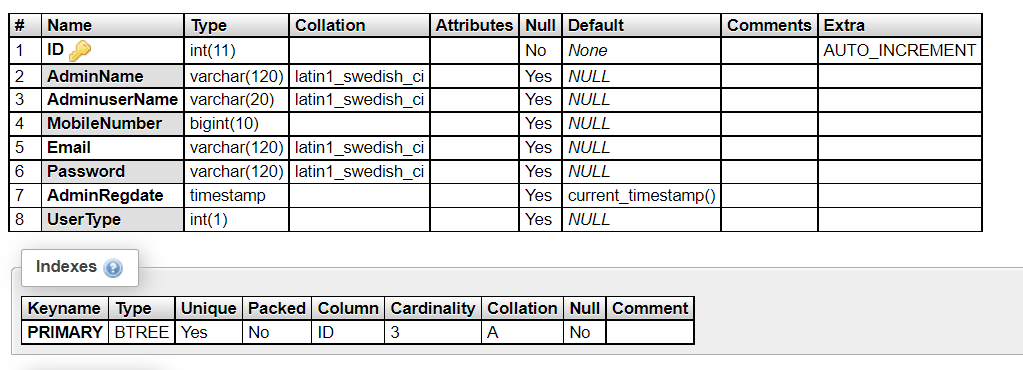
**Update Profile**

**Change Password**

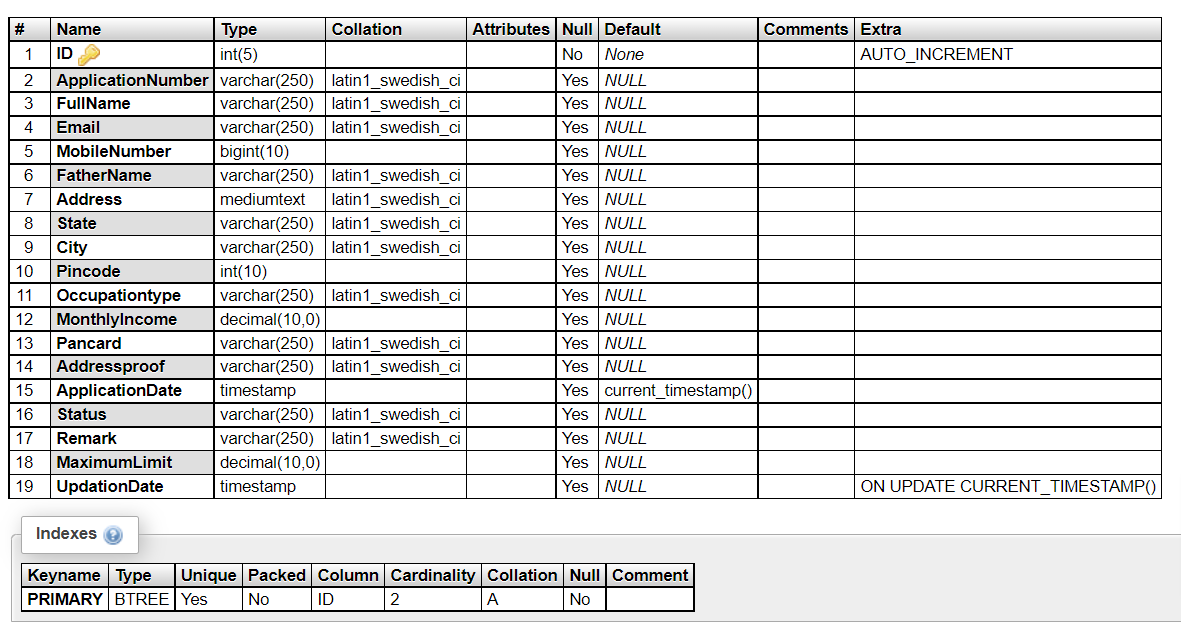
**MySQL Data Tables**

**Admin Table :**(Table name is admin)

This table stores admin / Sub-Admins personal and login details.

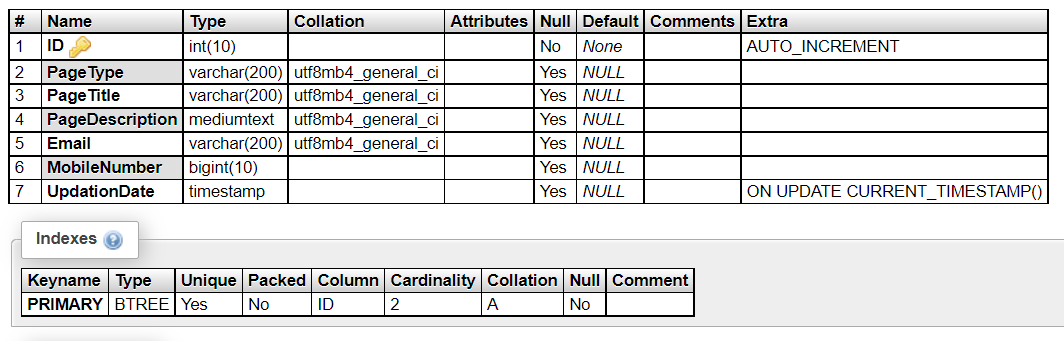
**Credit Card Application Table:** (Table name is tblapplication)

This table stores the details of credit card application which is received by users.



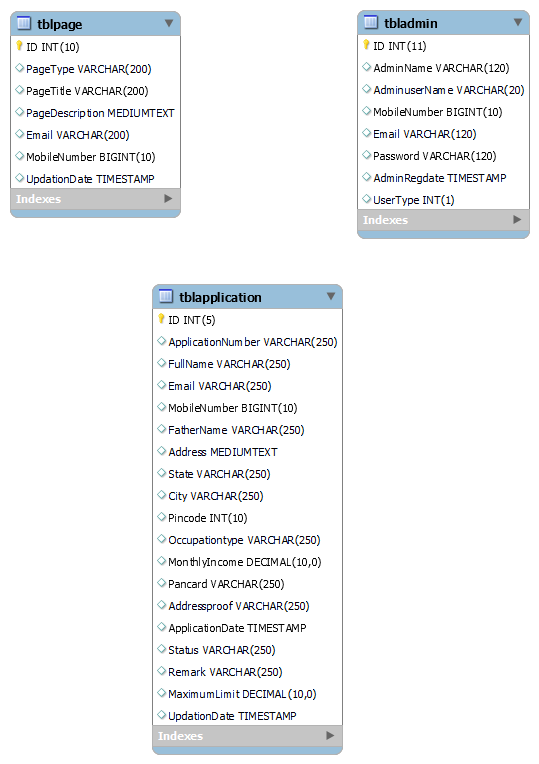
**Page Table:** (Table name is tblpage)

This table stores the content of about us and contact us page.



**Class Diagram:**

A description of set of objects that share the same attributes operations, relationships, and semantics



**Implementation and System Testing**

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

**System Testing**

The goal of the system testing process was to determine all faults in our project .The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2. Integration testing

**UNIT TESTING**

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require

* The procedures belonging to other units that the unit under test calls
* Non local data structures that module accesses
* A procedure to call the functions of the unit under test with appropriate parameters.

**1. Test for the admin module**

* **Testing admin login form-**This form is used for log in of administrator of the system. In this form we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.
* **Report Generation:** admin can generate report from the main database.

**INTEGRATION TESTING**

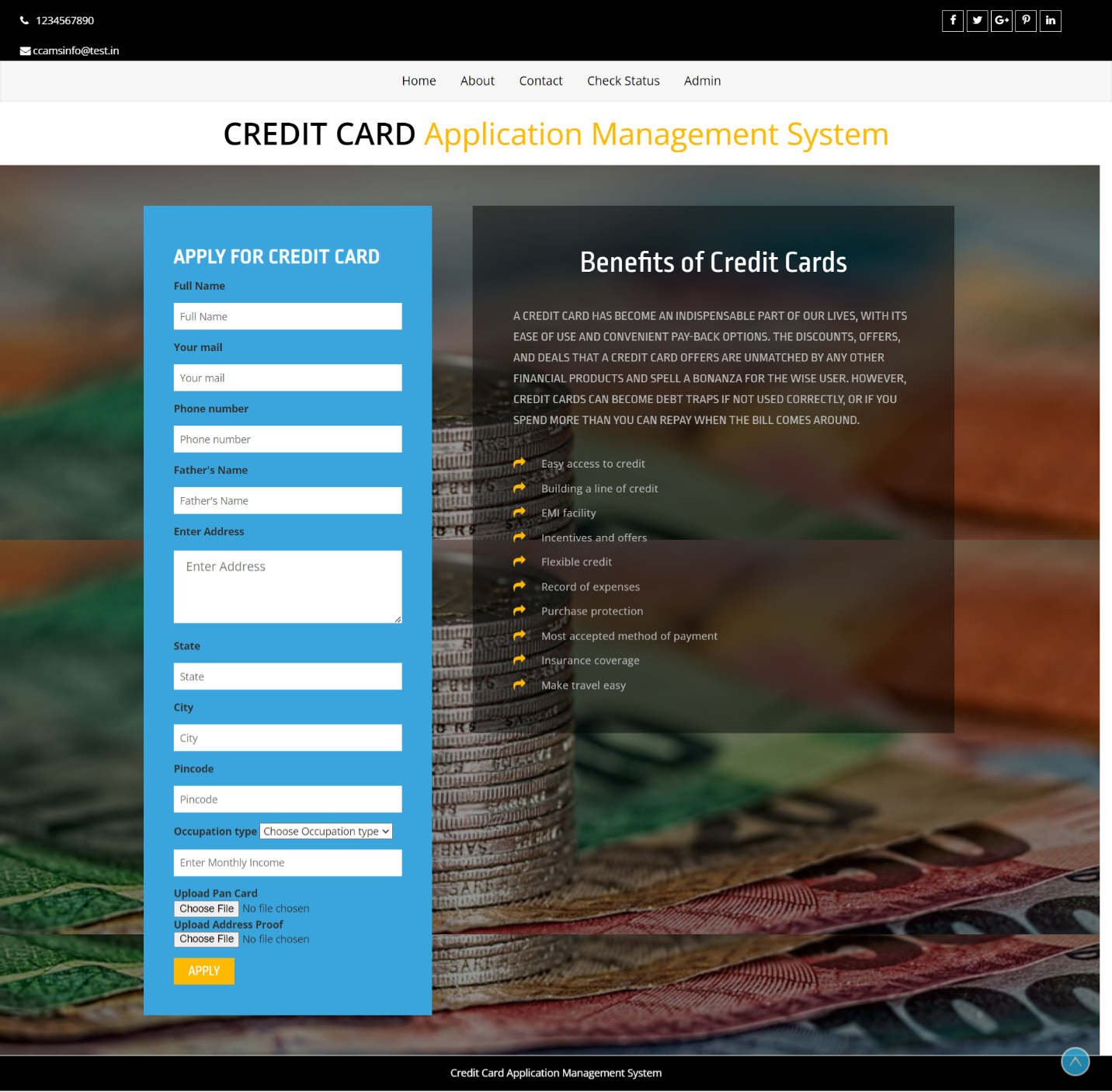
In the Integration testing we test various combination of the project module by providing the input.

The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

**EVALUATION**

**Project URL: http://localhost/ccams**

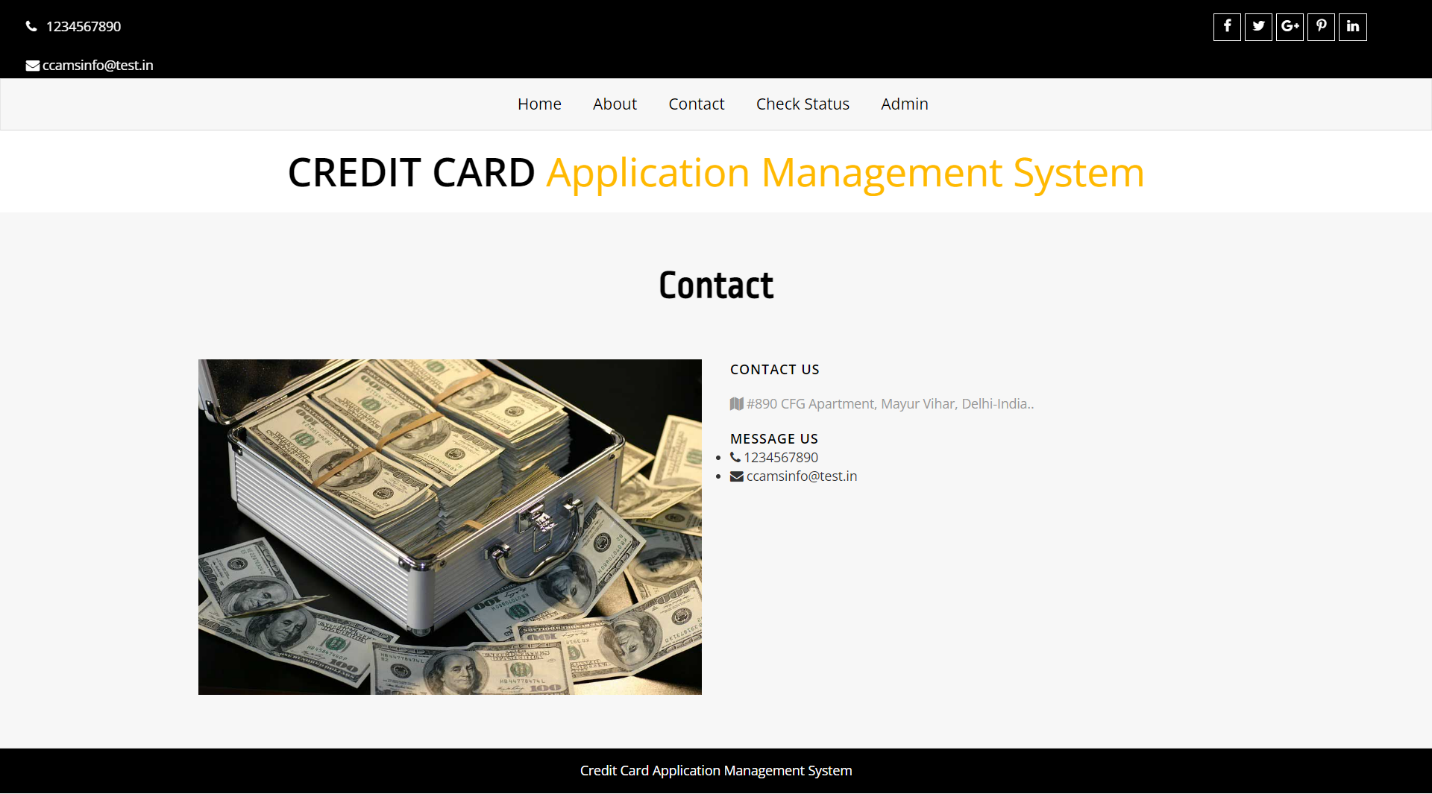
**Home Page**



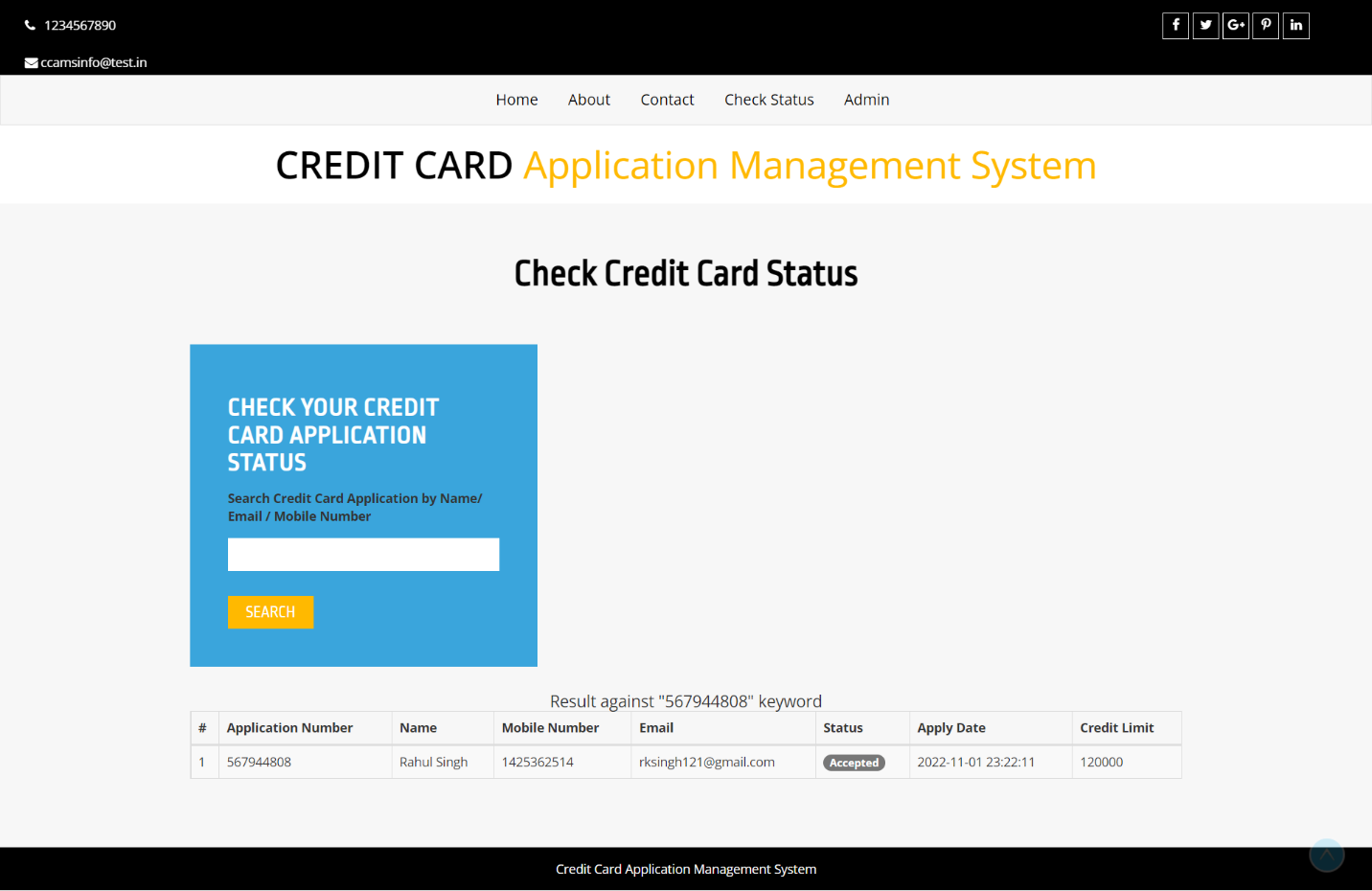
**About Us Page**



**Contact us**

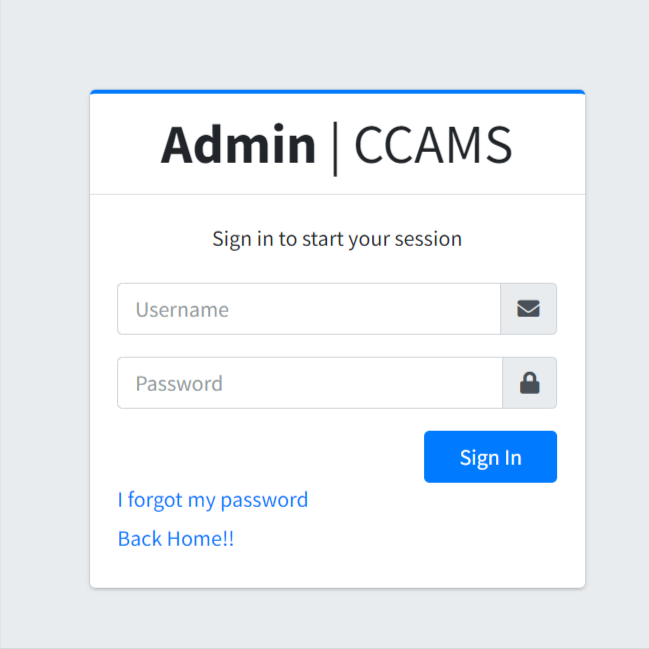


**Check Credit Card Status**



**Admin Panel / Sub-Admin**

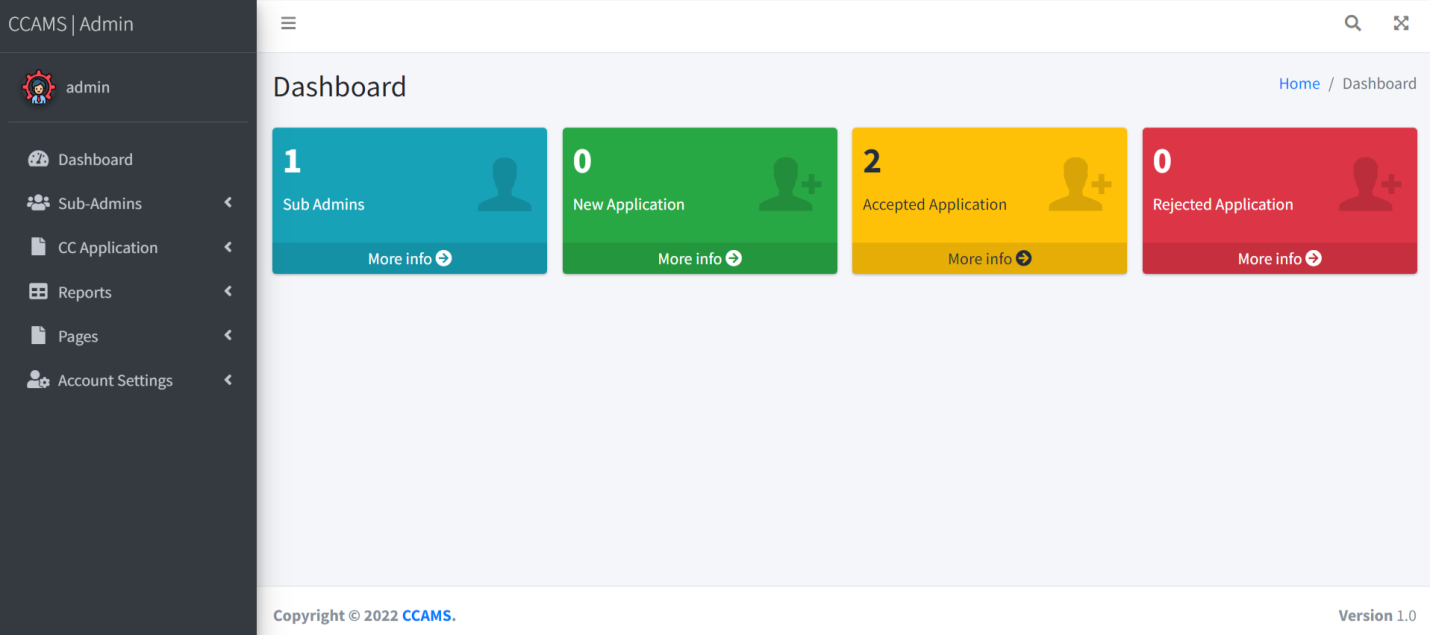
**Login Page**

****

**Forgot Password**

****

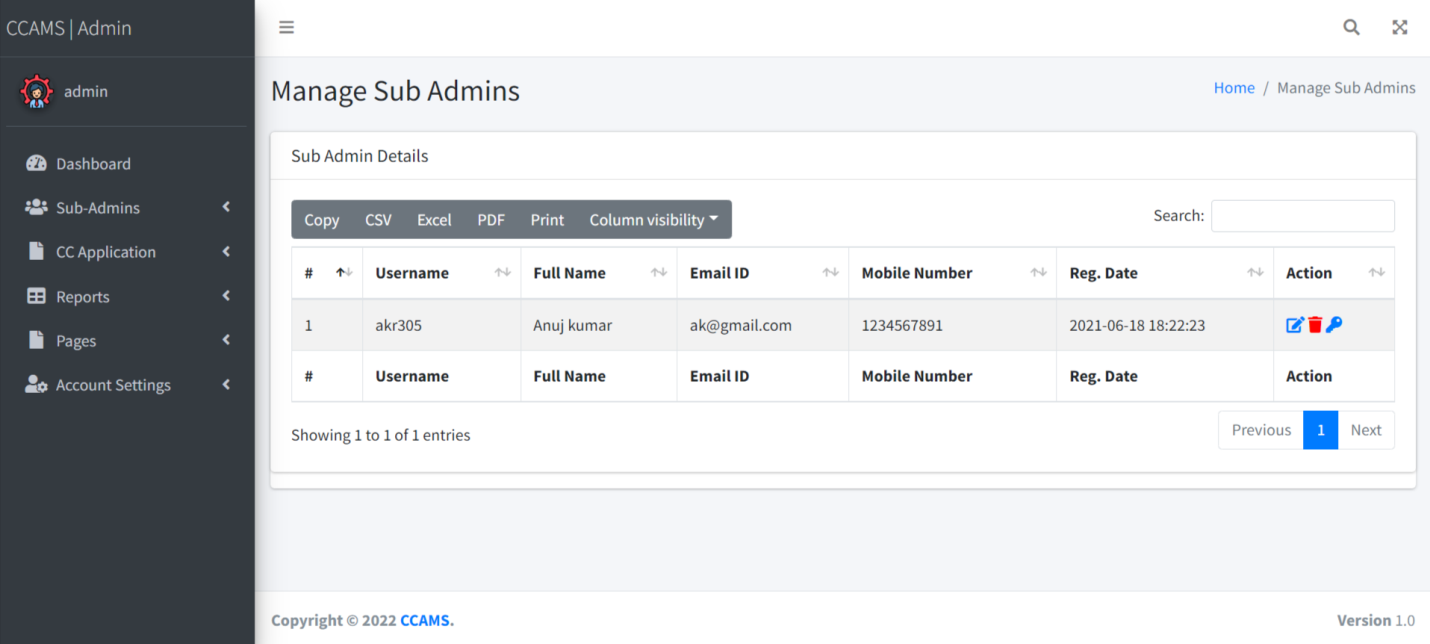
**Dashboard**

****

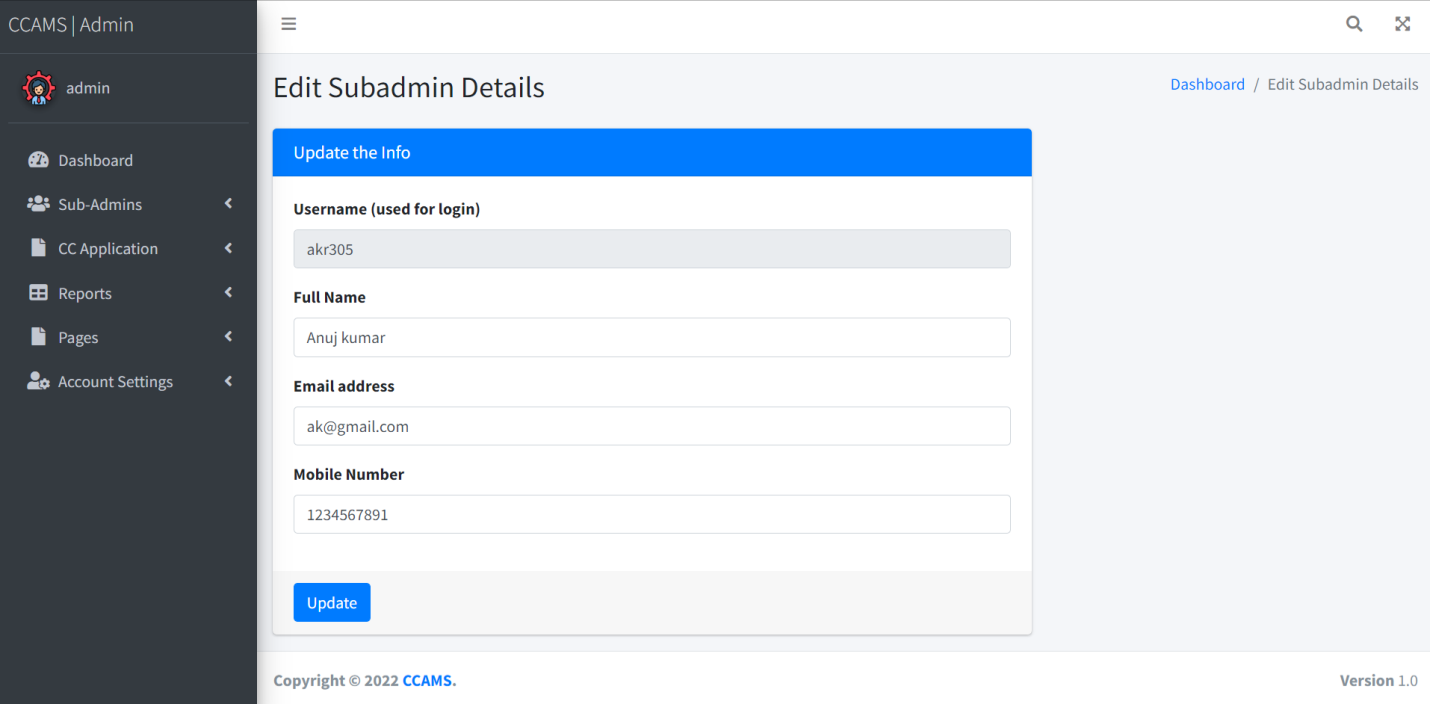
**Add Subadmin**

****

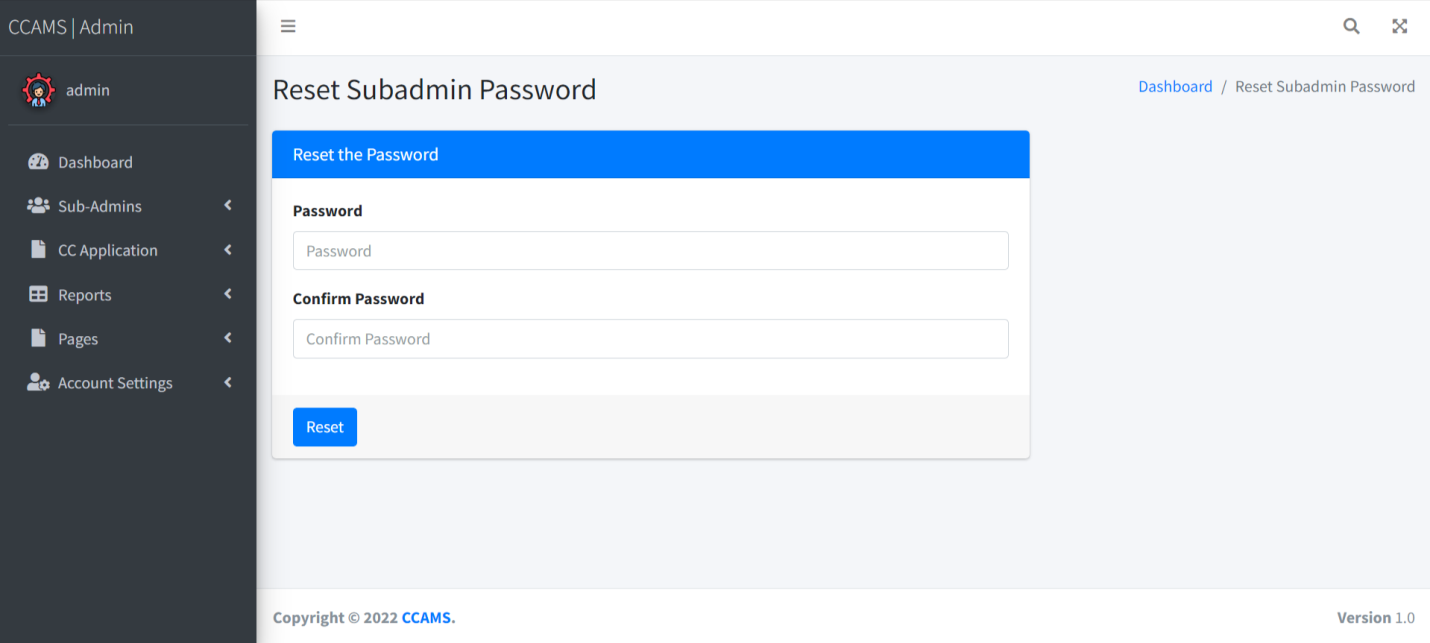
**Manage Subadmin**

****

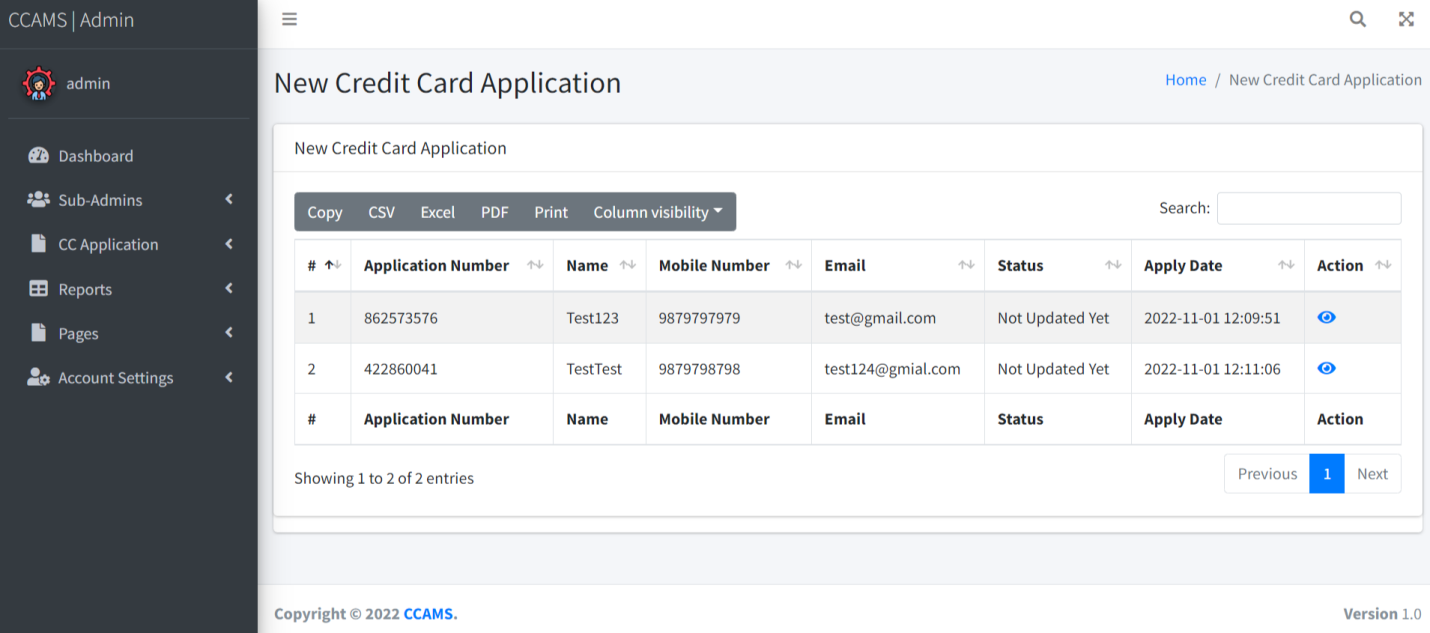
**Update Subadmin**

****

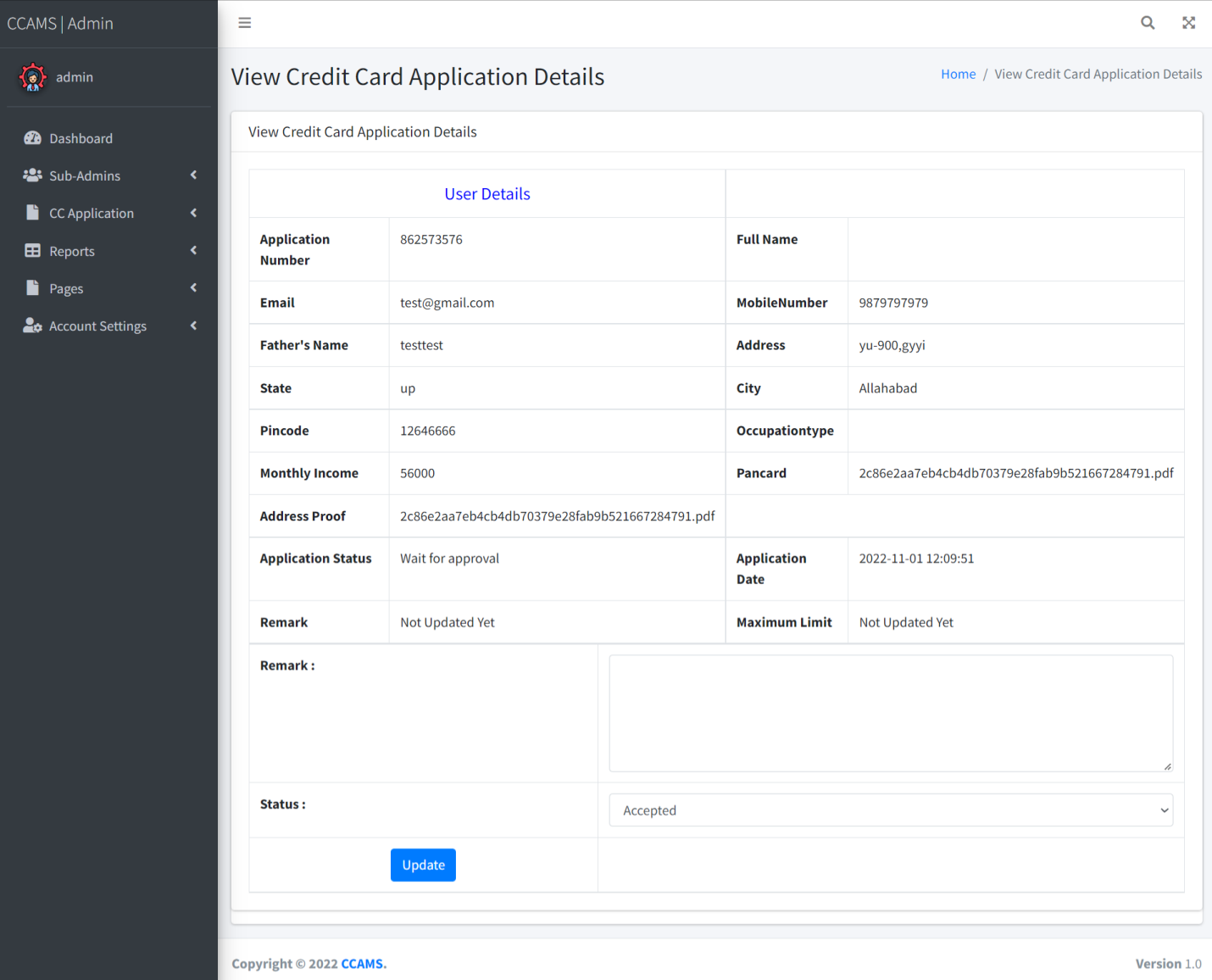
**Reset Subadmin Password**

****

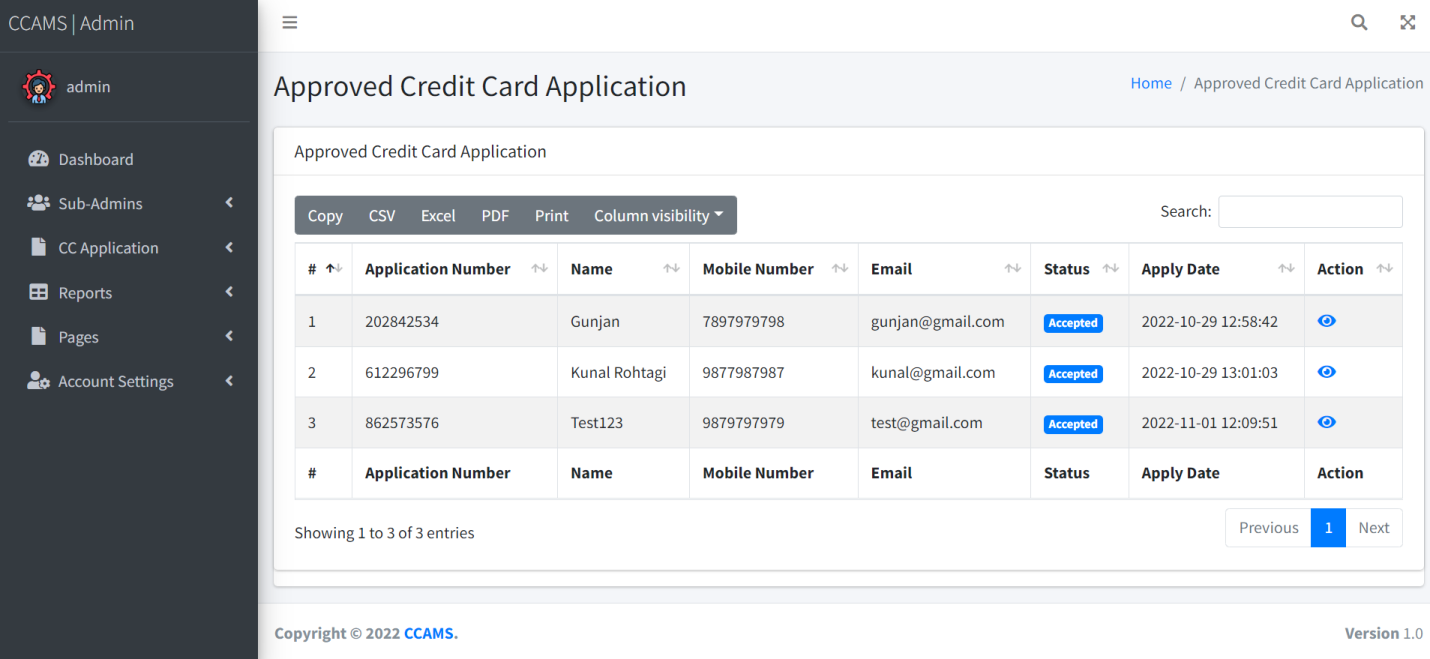
**New Application Request**

****

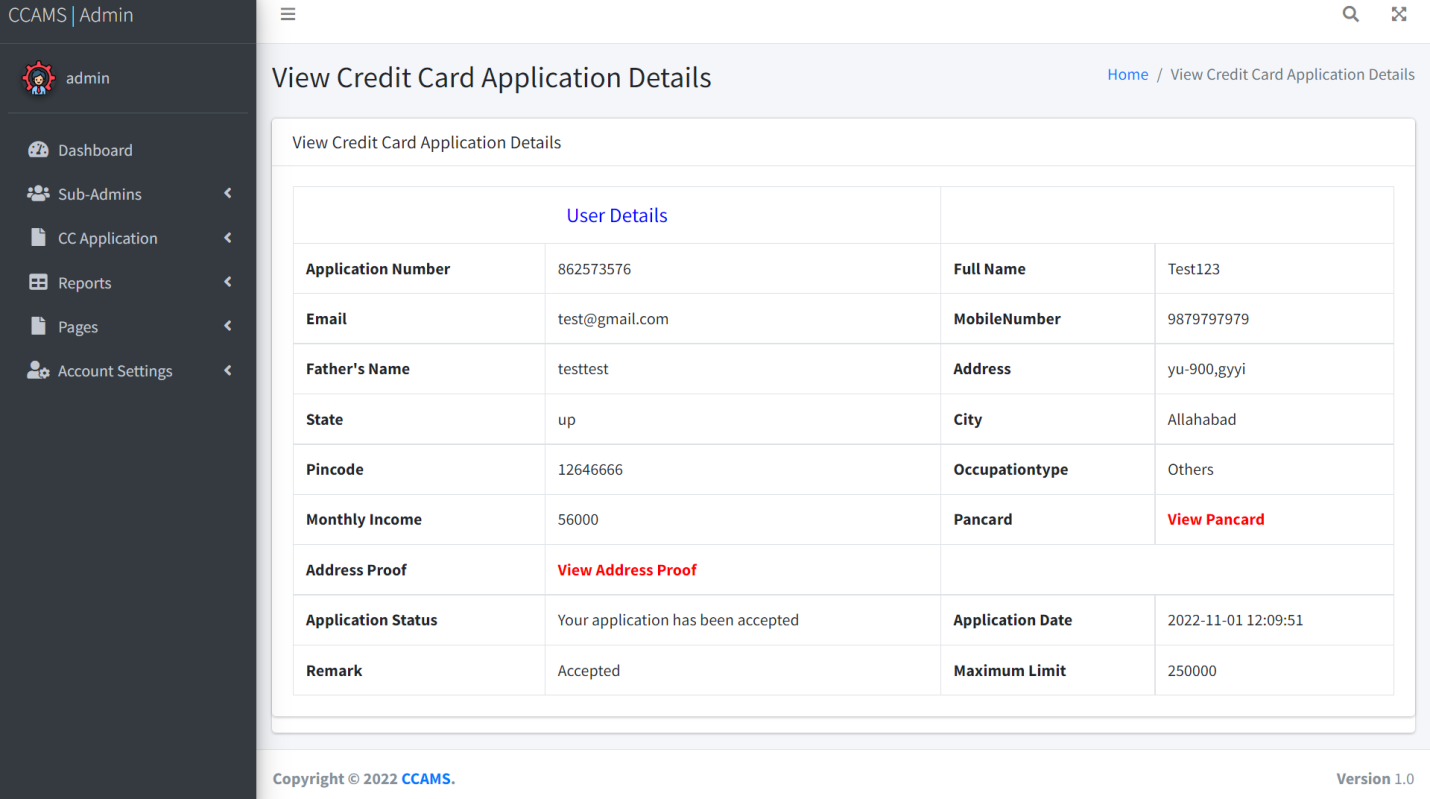
**View New Application Request Details**

****

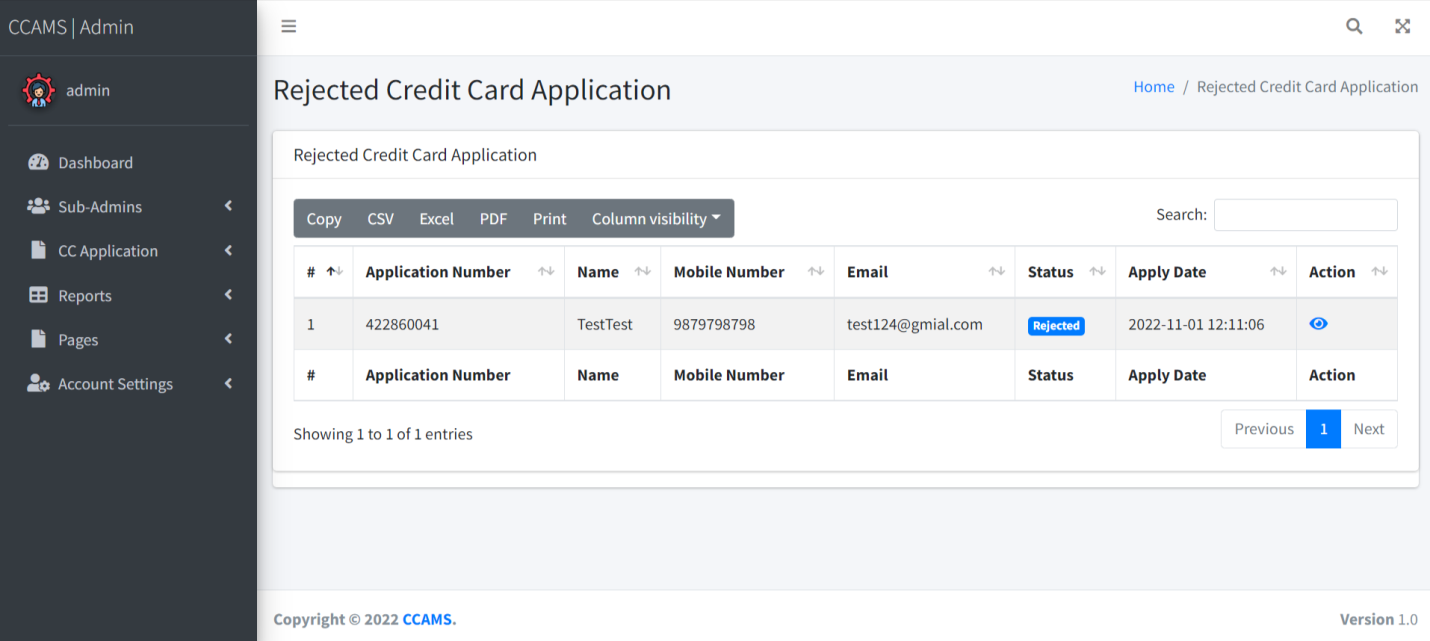
**Approved Application Request**

****

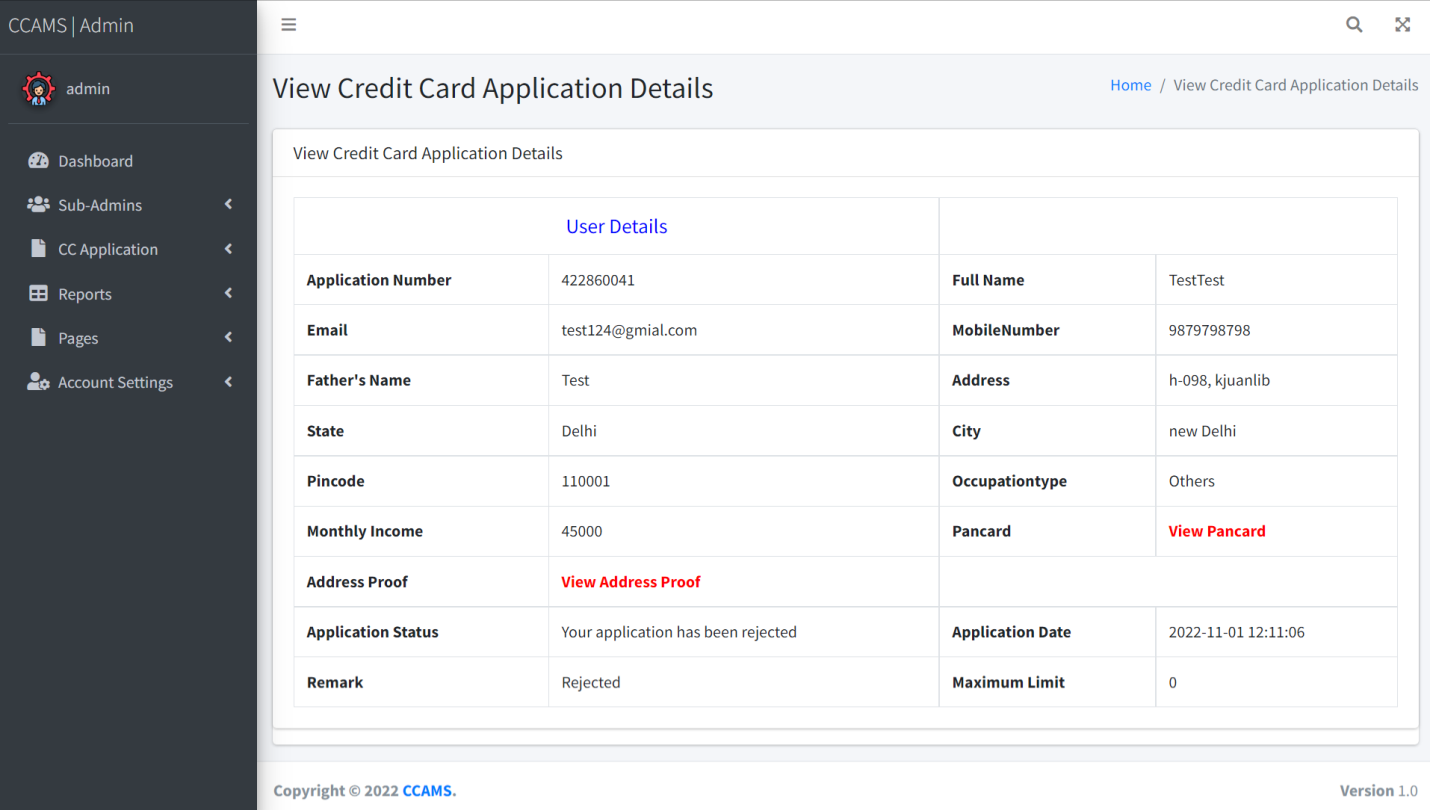
**View Approved Application Request Details**

****

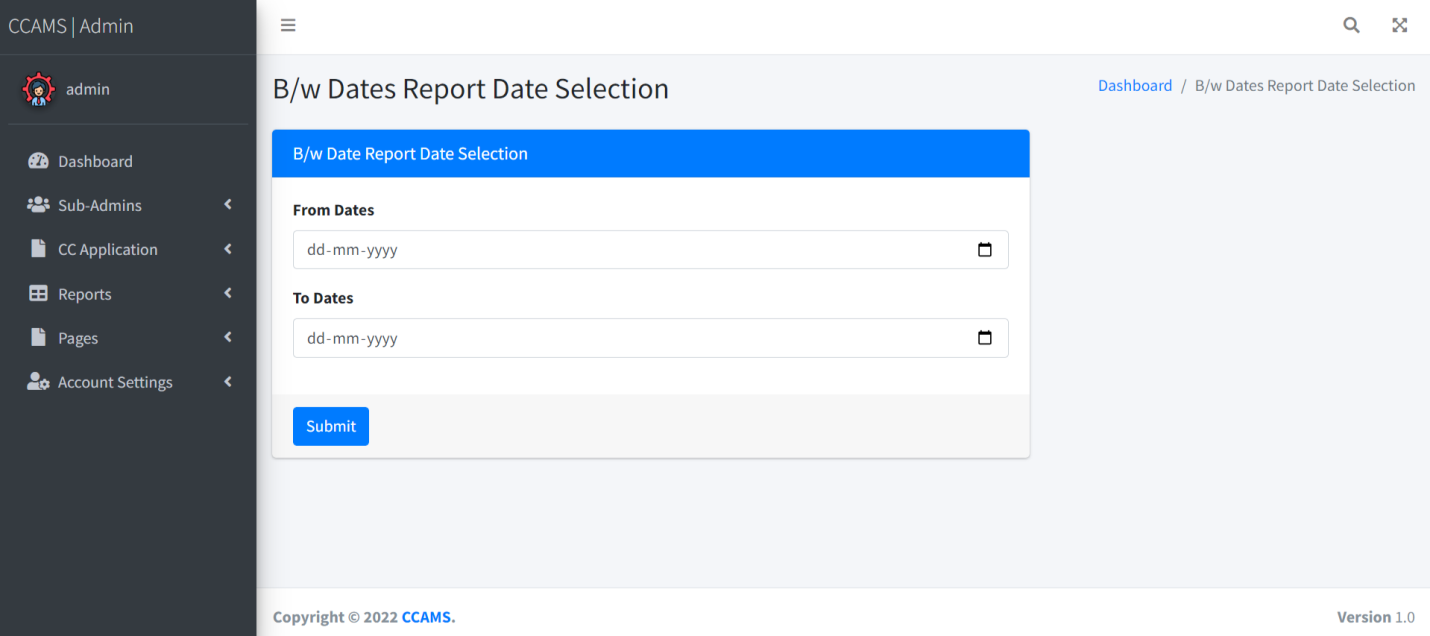
**Rejected Application Request**

****

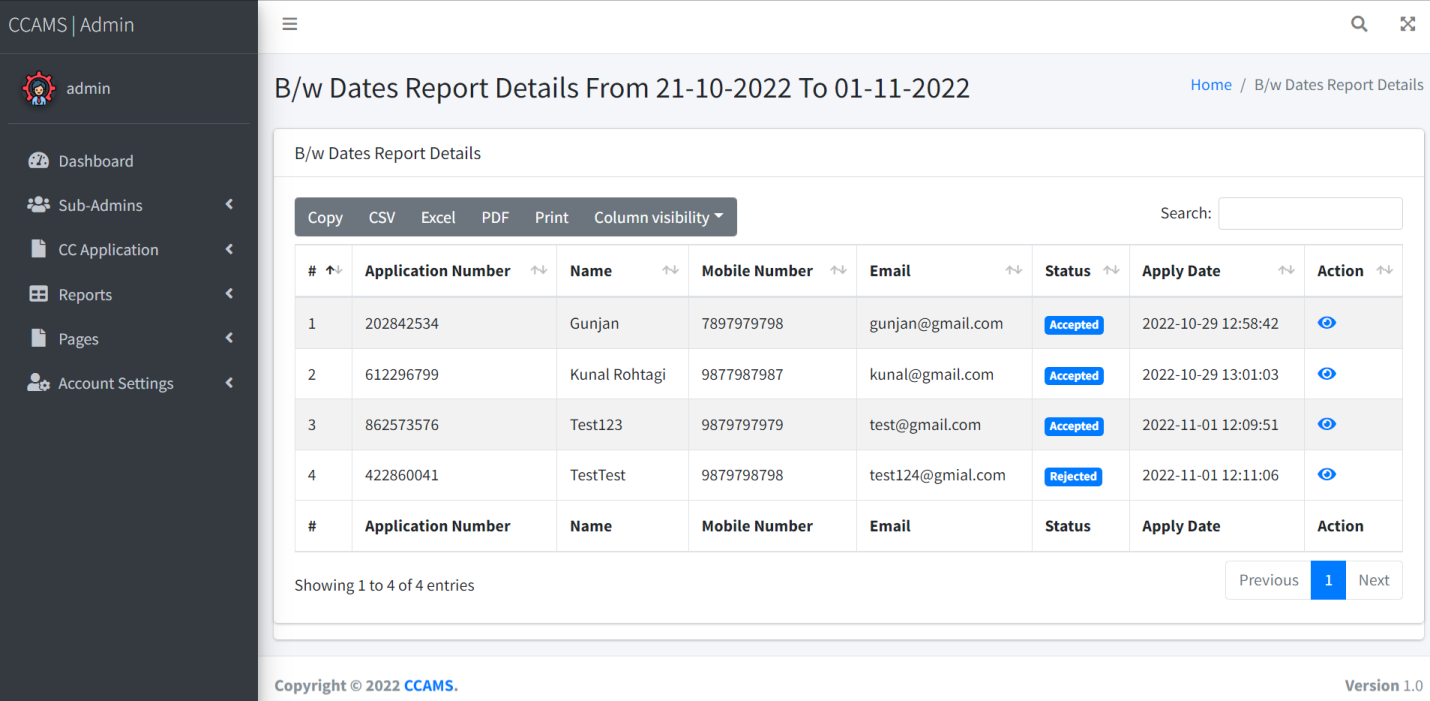
**View Rejected Application Request Details**

****

**Between Dates Reports**

****

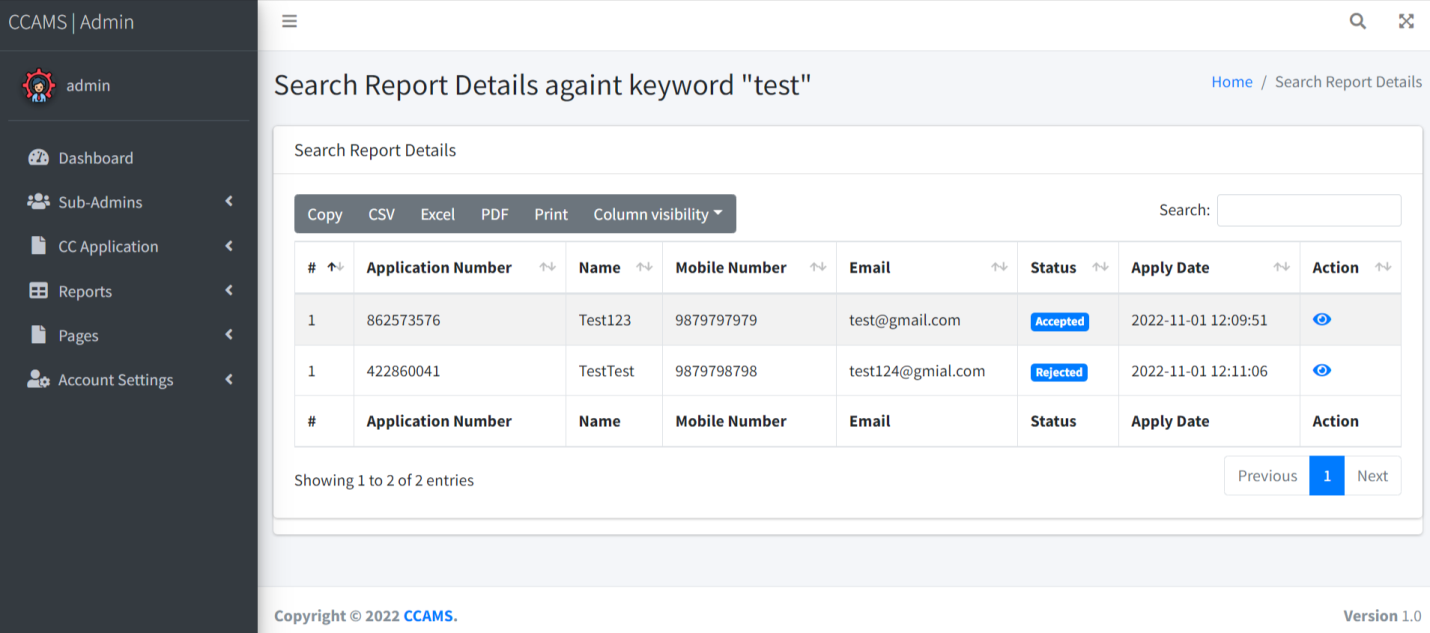
**View Details of Between Dates Reports**

****

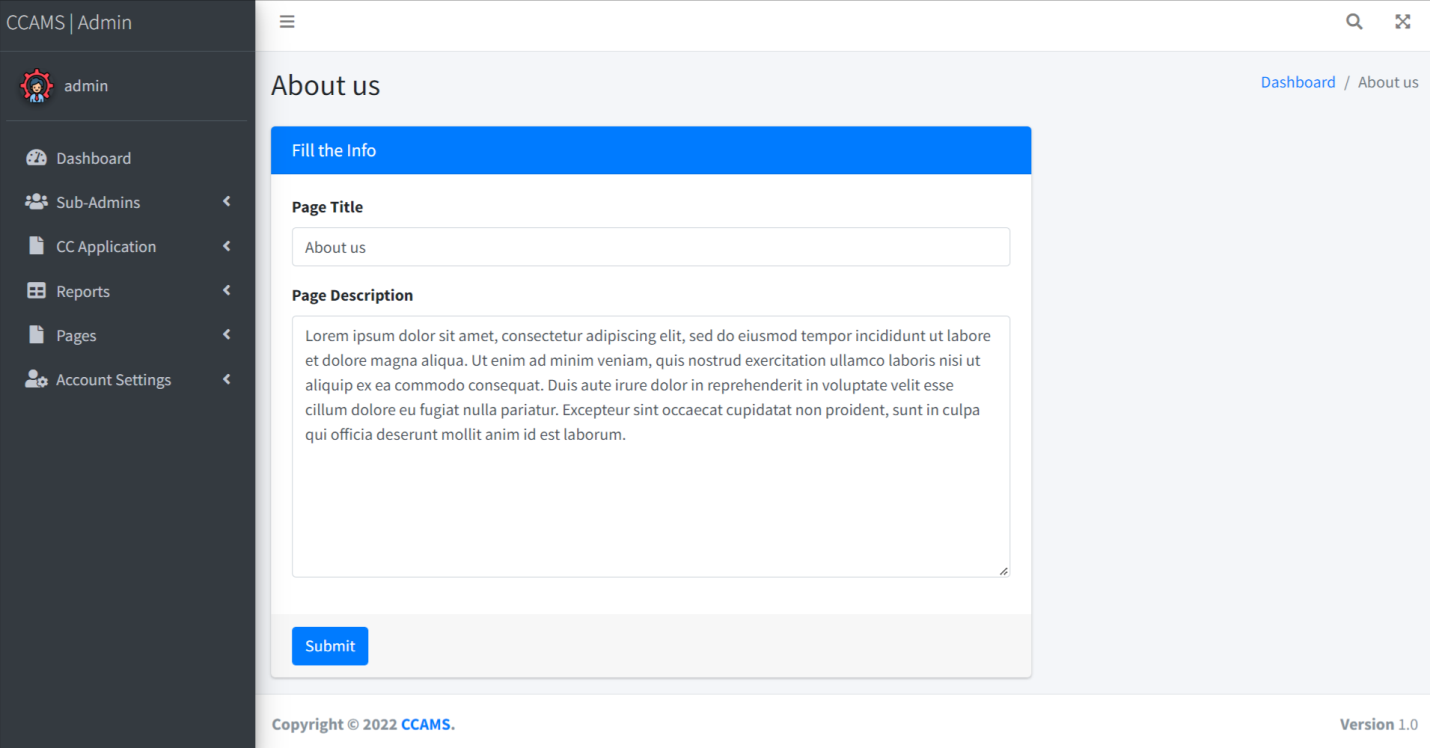
**Search Report**

****

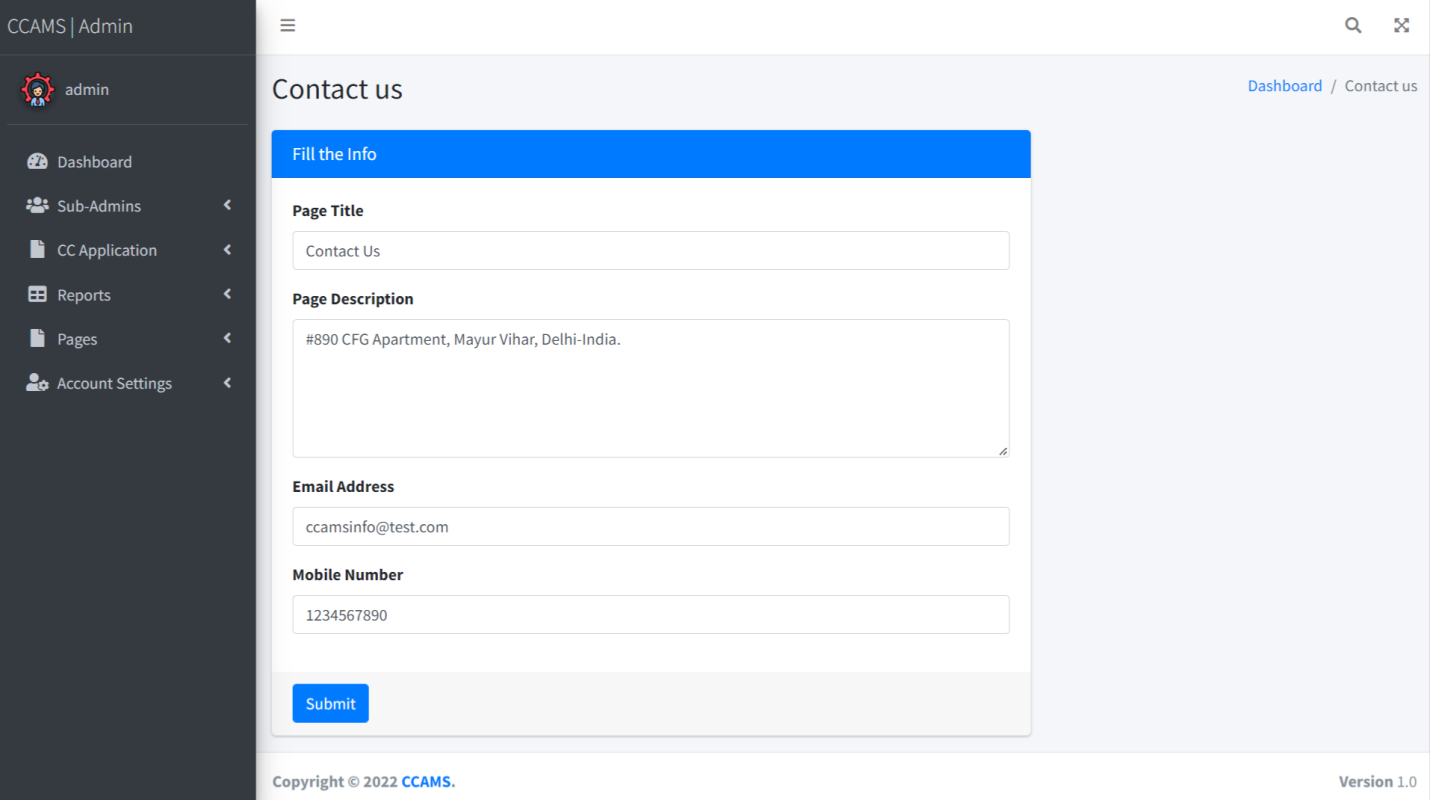
**View Search Report**

****

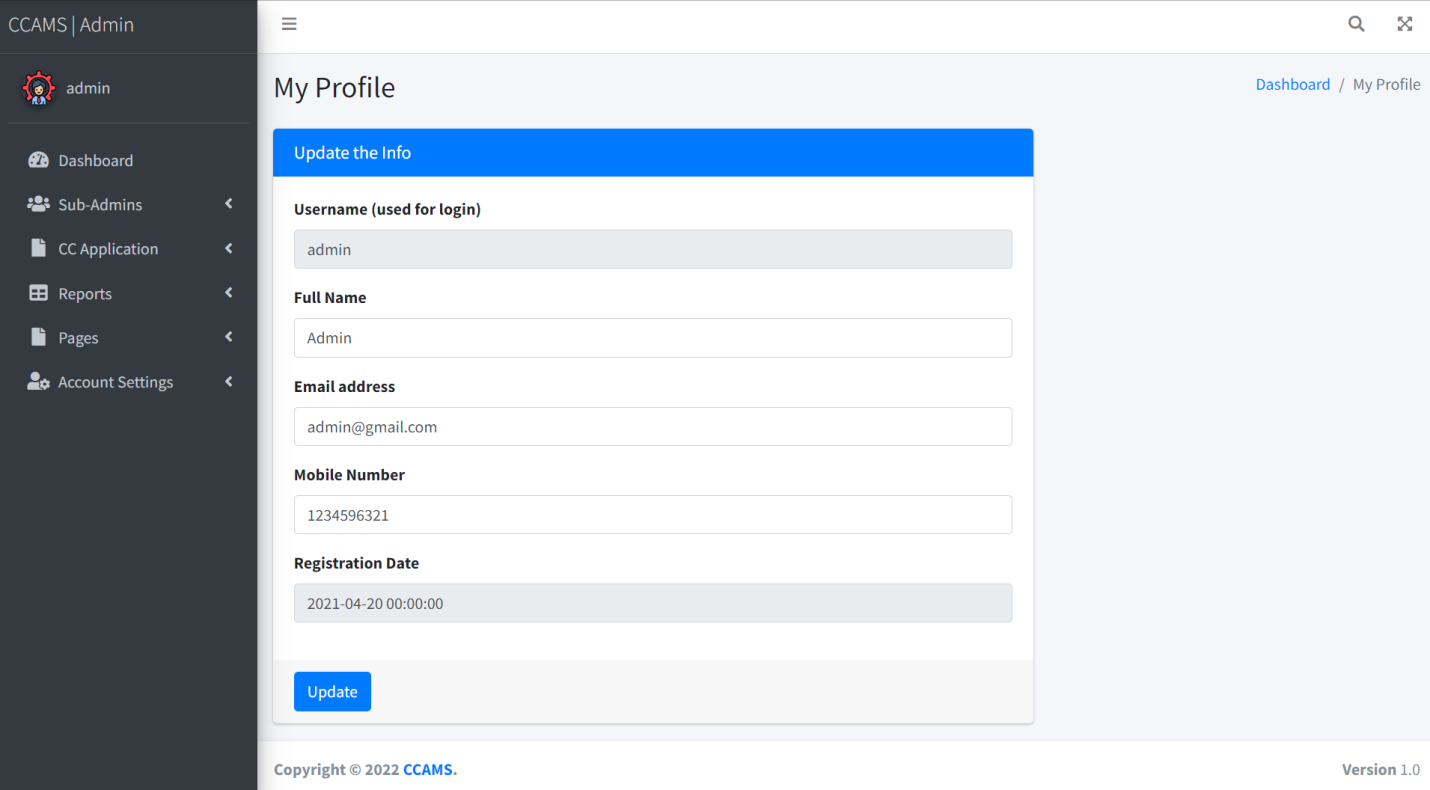
**About Us Page**

****

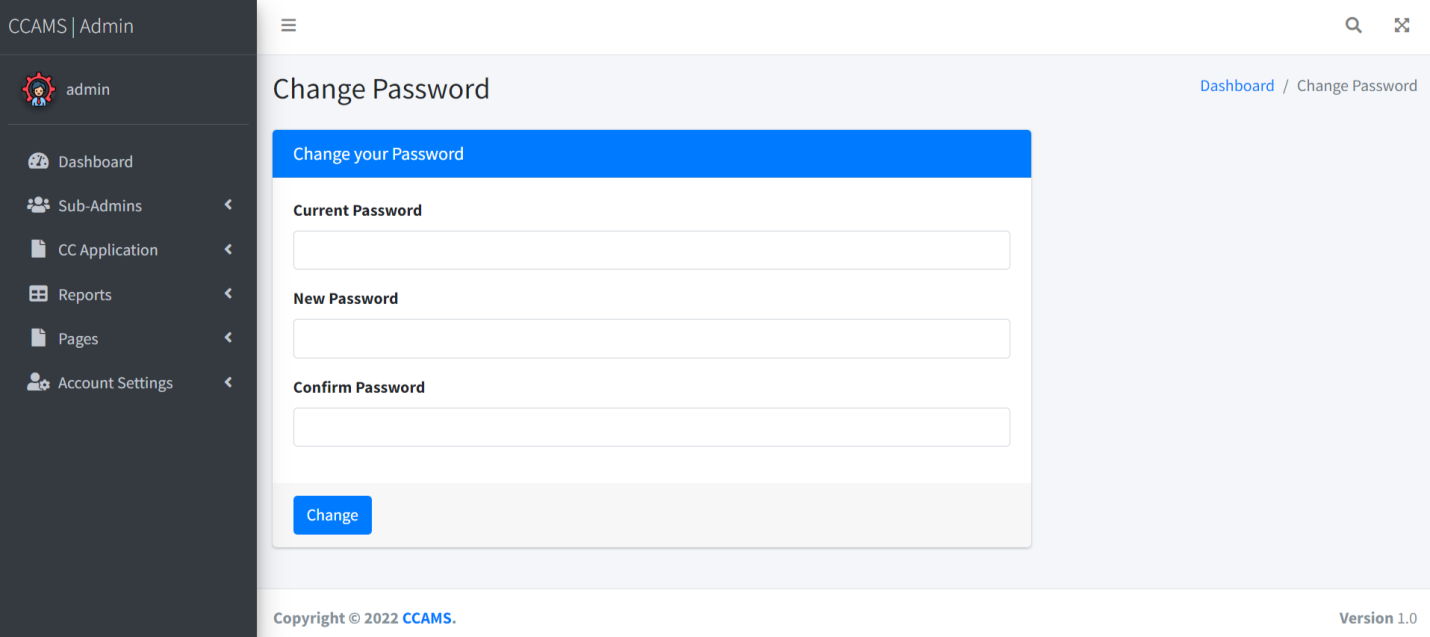
**Contact Us Page**

****

**Admin Profile**

****

**Change Password**

****

**CONCLUSION:**

This Application provides an online version of Credit Card Application Management System which will benefit the bank firm who want to maintain records of credit card holder and also help to customer to apply credit card online without wasting a time and apply with their convenience.

It makes entire process online and can generate reports.

The Application was designed in such a way that future changes can be done easily. The following conclusions can be deduced from the development of the project.

* Automation of the entire system improves the productivity.
* It provides a friendly graphical user interface which proves to be better when compared to the existing system.
* It gives appropriate access to the authorized users depending on their permissions.
* It effectively overcomes the delay in communications.
* Updating of information becomes so easier.
* System security, data security and reliability are the striking features.
* The System has adequate scope for modification in future if it is necessary.

**References**

**For PHP**

* <https://www.w3schools.com/php/default.asp>
* <https://www.sitepoint.com/php/>
* <https://www.php.net/>

**For MySQL**

* <https://www.mysql.com/>
* [http://www.mysqltutorial.org](http://www.mysqltutorial.org/)

**For XAMPP**

* <https://www.apachefriends.org/download.html>

