**Project Report**

**on**

**Online Student Registration System**

**Bachelor of Computer Applications**

**(Session 2020 - 2021)**

Submitted by

Ashif Ali 1319075

Anup Kumar Sah 1319073

Arjun Kumar 1319081

Kishan Kumar 1319092

Project In Charge

Dr. Dalip Kamboj

Professor (Assistant),



**M.M. INSTITUTE OF COMPUTER TECHNOLOGY & BUSINESS MANAGEMENT**

**MAHARISHI MARKANDESHWAR (DEEMED TO BE UNIVERSITY)**

**MULLANA–AMBALA, HARYANA (INDIA) - 133207**

**(Established under Section 3 of the UGC Act, 1956)**

**(Accredited by NAAC with Grade ‘A’)**

Table of Contents

1. Abstract
2. List of diagrams
   1. Data Flow Diagram
   2. ER Diagram
3. List of tables
4. Student Table
5. Admin Table
6. Admission Table
7. Course Table
8. Specialization Table
9. Admin Table
10. Introduction
11. Requirement Analysis
12. Hardware requirement specification:
13. Software requirement specification:
14. Software Design
    1. DFD (Data Flow Diagram)
    2. ER Diagram
    3. Database Design
    4. Designing Form
15. Coding
16. Testing
17. Limitation
18. Scope of the project
19. Conclusion
20. Bibliography
21. **Abstract**

Student Online admission is a vital part one of any university’s running because students are what keep a University alive. Currently the student submits hardcopy of filled application form to the college and office staff enters all data into excel file and write same in manual register.

Proposed Online Student Registration System will eliminate all the manual intervention and increase the speed of whole process

1. **List of Diagrams**
   1. **Data Flow Diagram**
   2. **ER Diagram**
2. **List of Tables**
   1. **Student Table**
   2. **Admin Table**
   3. **Admission Table**
   4. **Course Table**
   5. **Specialization Table**
   6. **Admin Table**
3. **Introduction**

Student Online admission is a vital part one of any university’s running because students are what keep a University alive. The Student admission is one of the most important activities within a university as one cannot survive without student. A poor admission system can mean fewer students being admitted into a university because of mistakes or an overly slow response time

Currently the student submits hardcopy of filled application form to the college and office staff enters all data into excel file and write same in manual register. College staff checks all the application and calculates marks list for selection process. After the manual work the college staff sends letter to selected candidates or they post in the notice board. The student has to come to the college to check the selected candidate’s lists in the notice board.

Proposed Online Student Registration System will eliminate all the manual intervention and increase the speed of whole process. This Web portal will help the student to get the information about a particular course and then they can easily register them self in a particular course. After successful submission, system will give unique registration no for each student. Student can login into system by using registration id and they can check shortlisted candidate details. The management of the college can easily see the records of the students, their registered course and fees. College staff can short list the candidates instantly without any errors.

Through this online system we overcome many Problems.

* Time and Money is saved.
* Nothing is done manually.
* Long lines in the department for these issues.

1. **Requirement Analysis**

**5.1 Hardware requirement specification:**

**Operating System:** Windows XP / Windows 7/ Windows 8

**Hard disk:** Minimum 40 GB

**RAM:** Minimum 512 MB

**Processor –** Minimum Pentium Dual Xenon Processor

Keyboard and Mouse

**5.2 Software requirement specification:**

XAMPP Software v 3.2.1

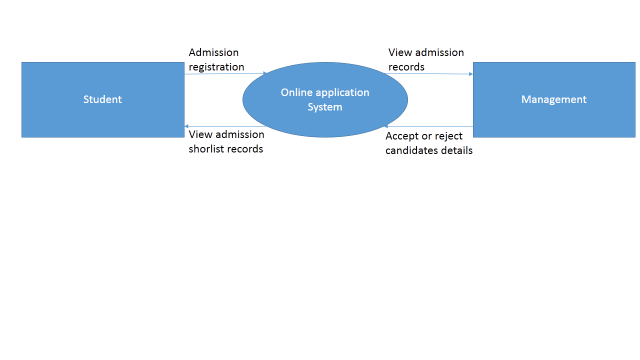
Apache server 1.8.2

MySQL database Server 5.5

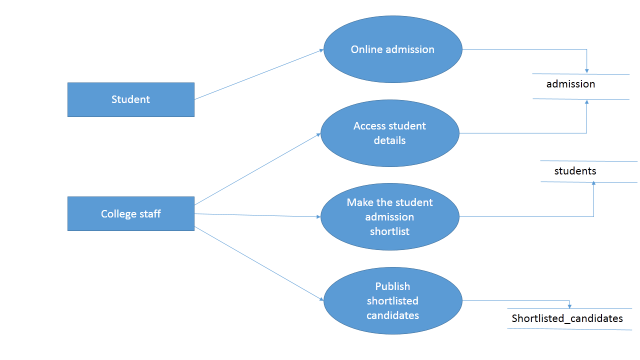
**IDE**: VS Code 1.53

1. **Software Design**

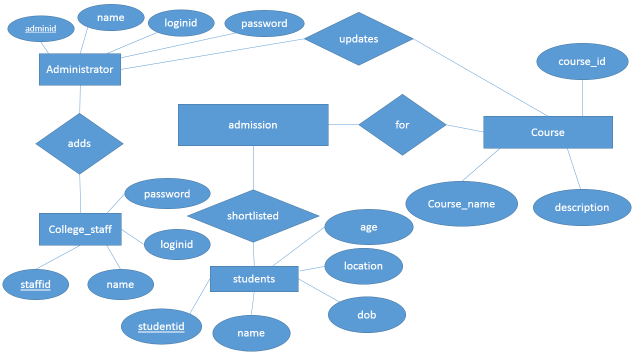
**6.1 DFD (Data flow Diagram):**

**Level 0:**

**Level 1:**

****

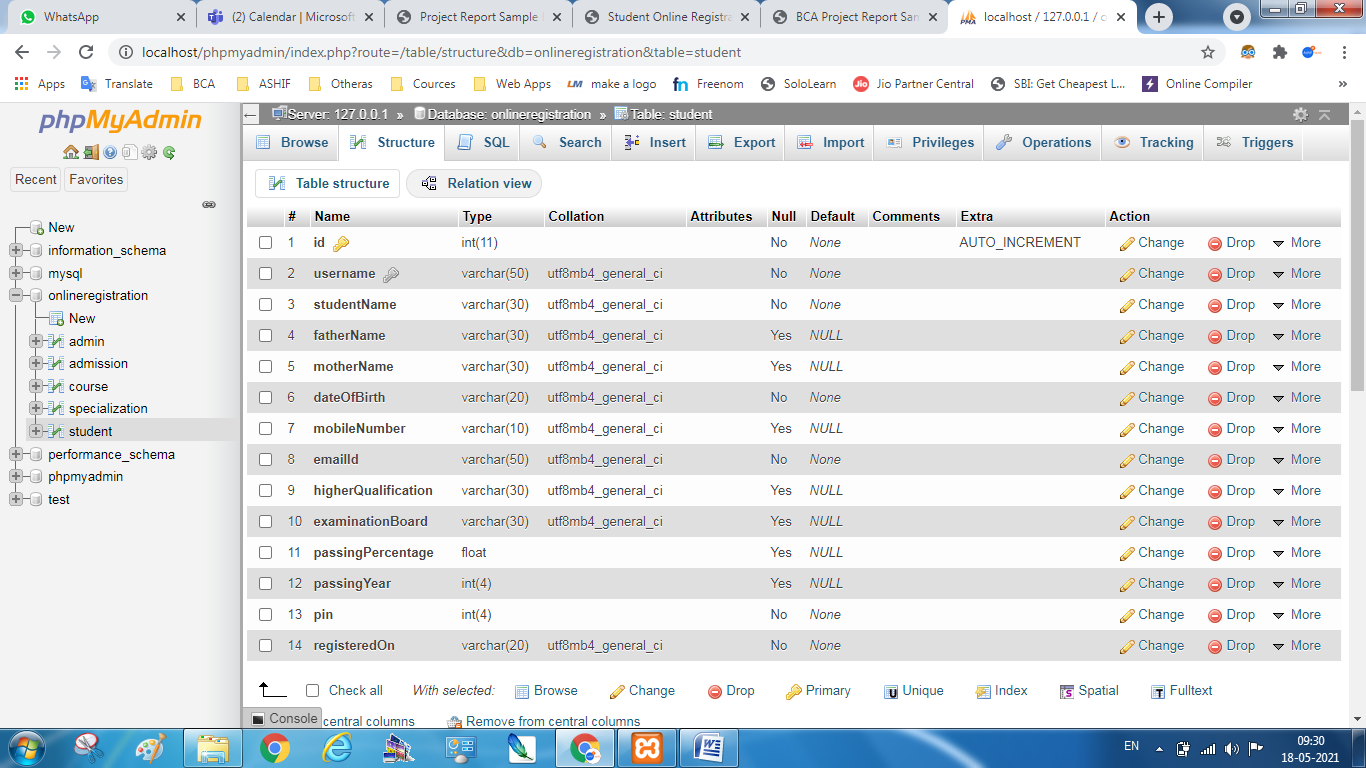
**6.2 ER Diagram:**

****

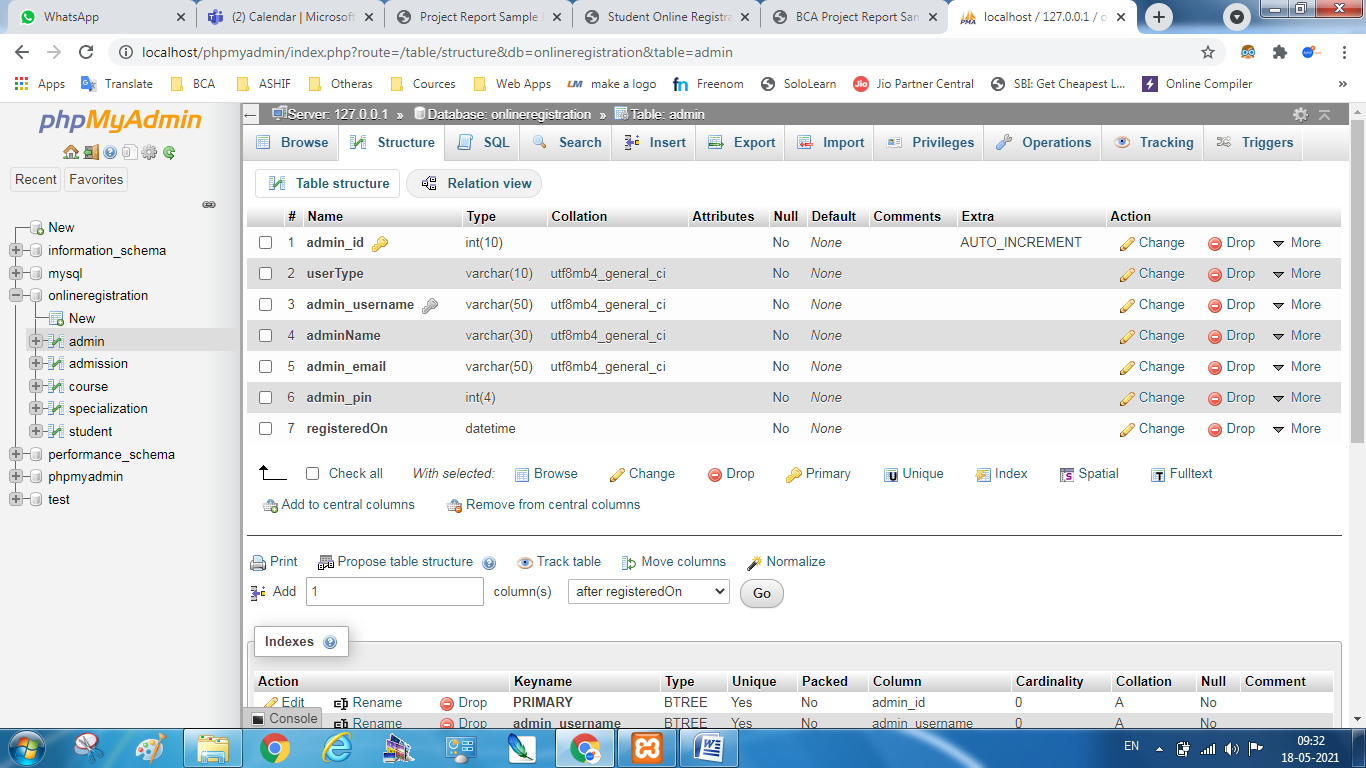
**6.3 Database Design**

**Table Structure**

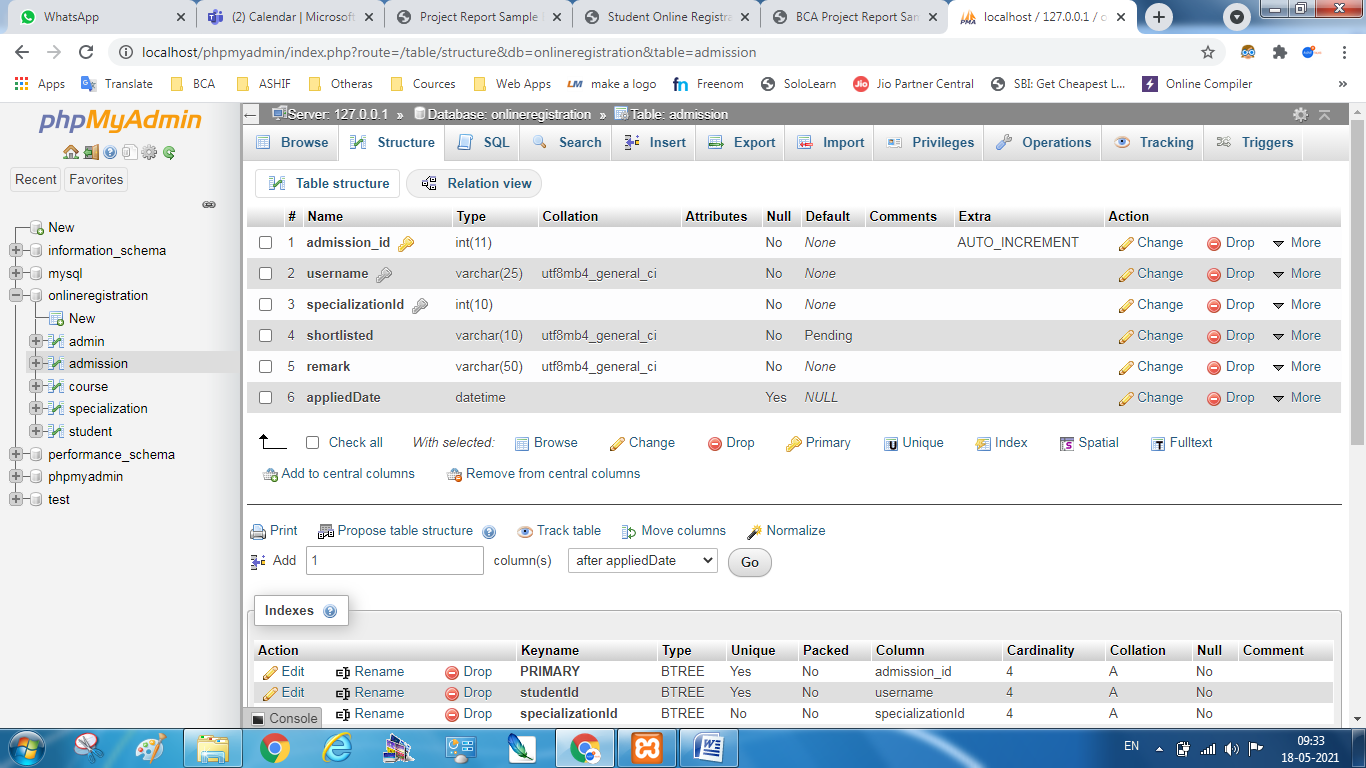
Student Table

****

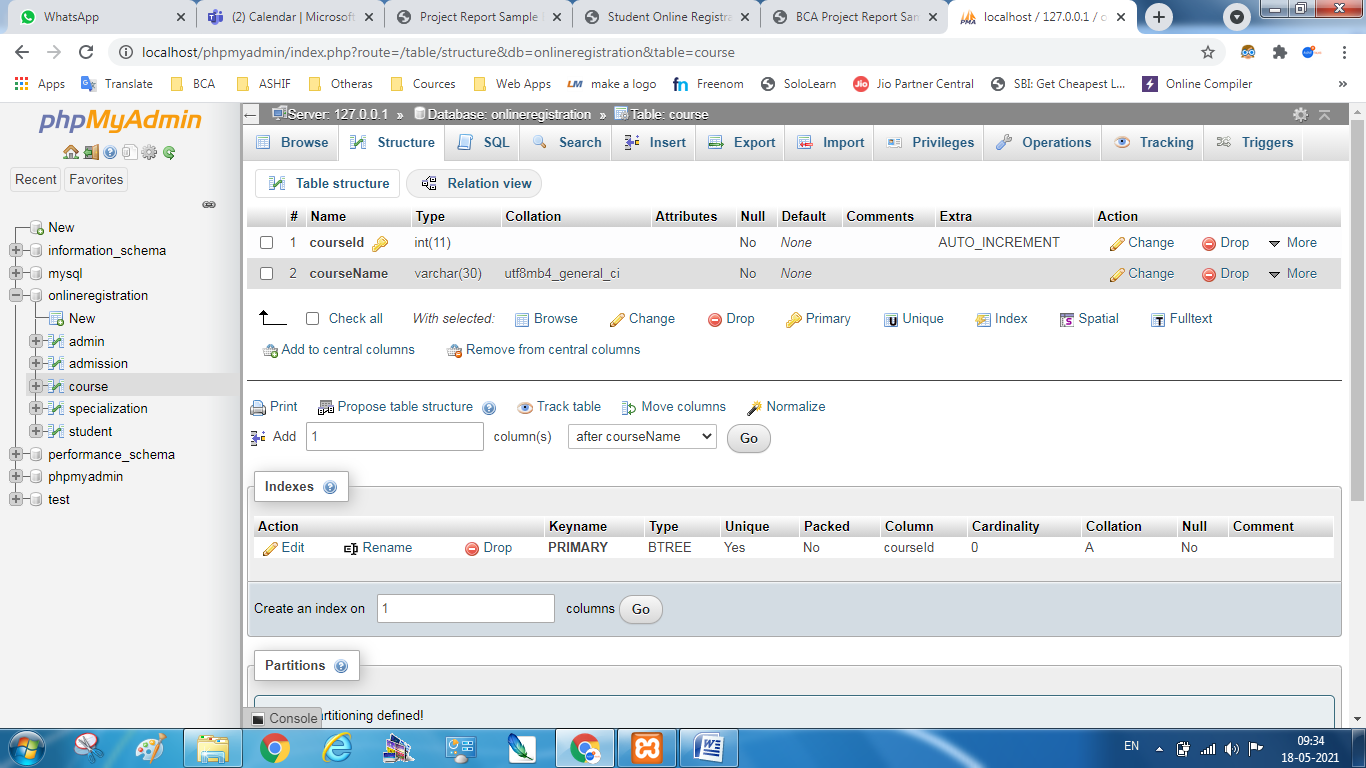
Admin Table



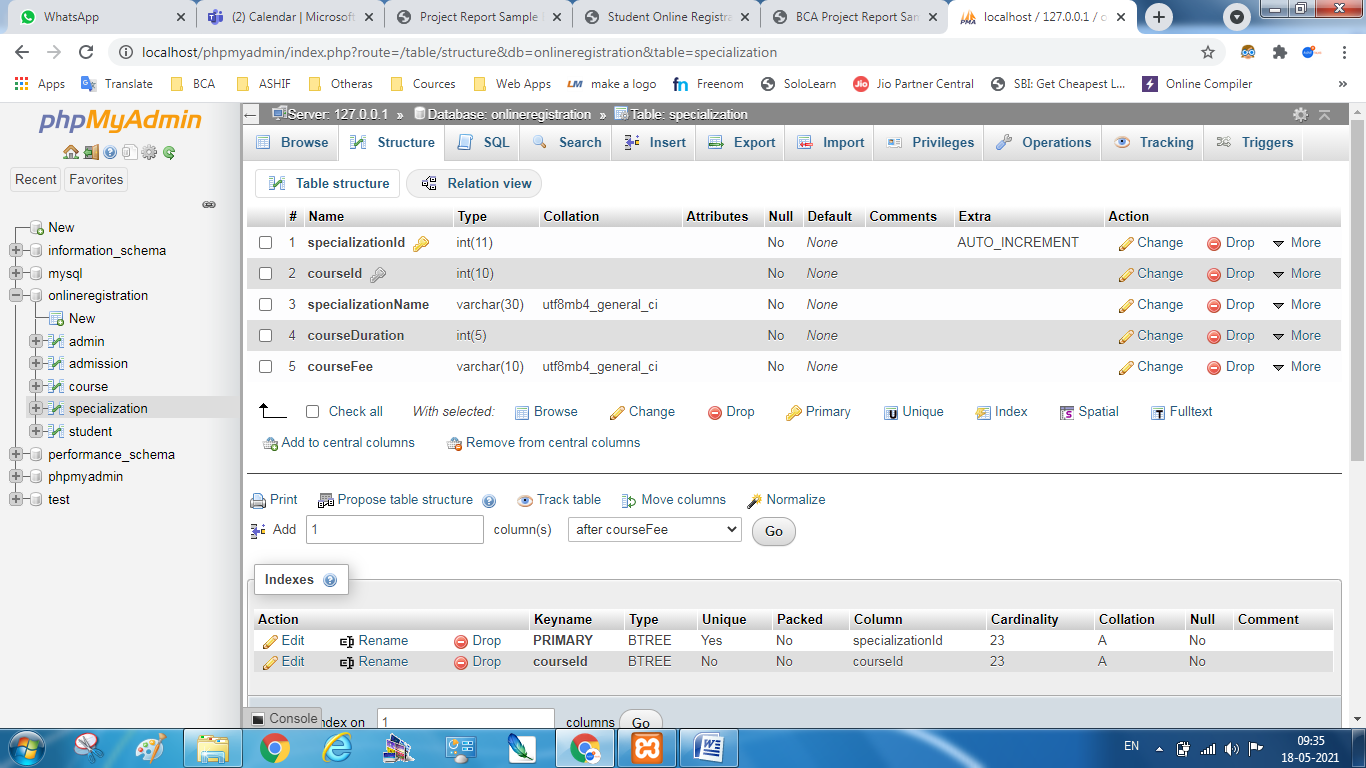
Admission Table



Course Table



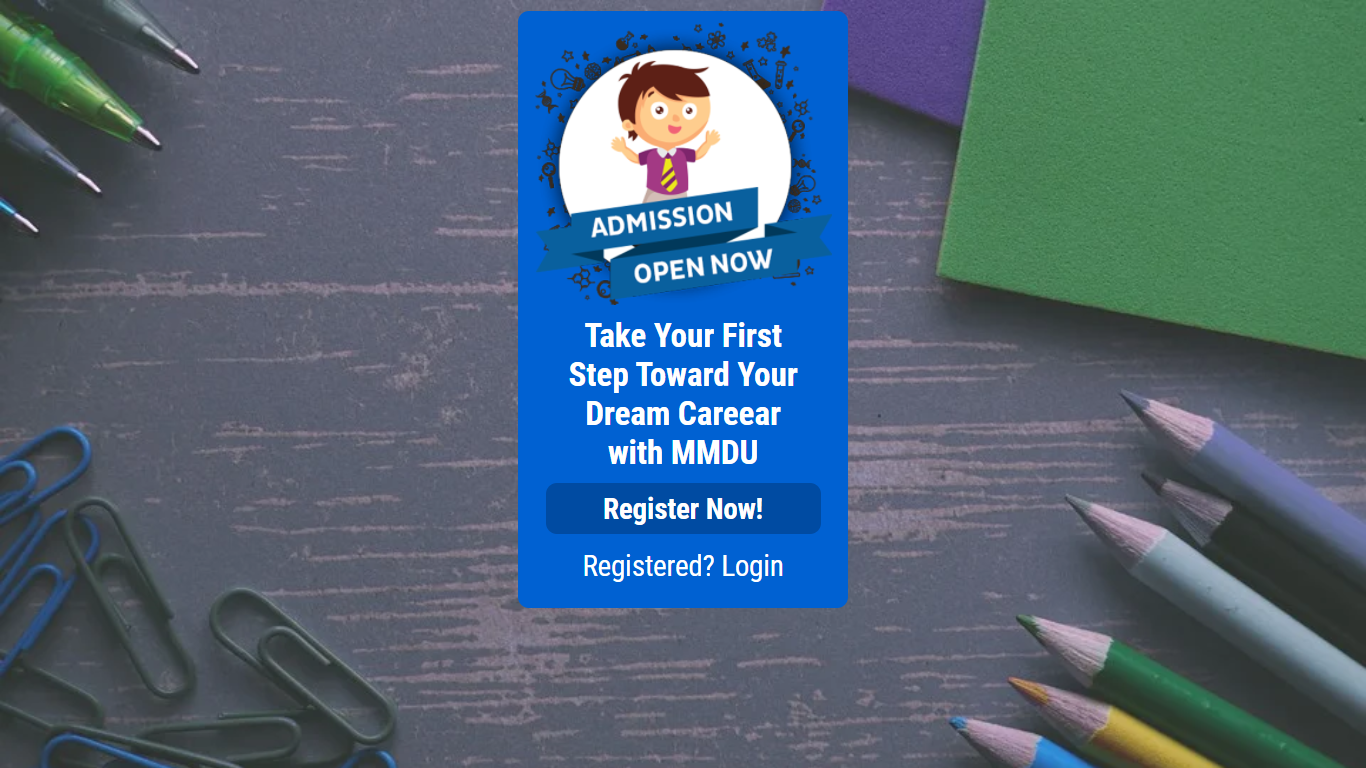
Specialization Table



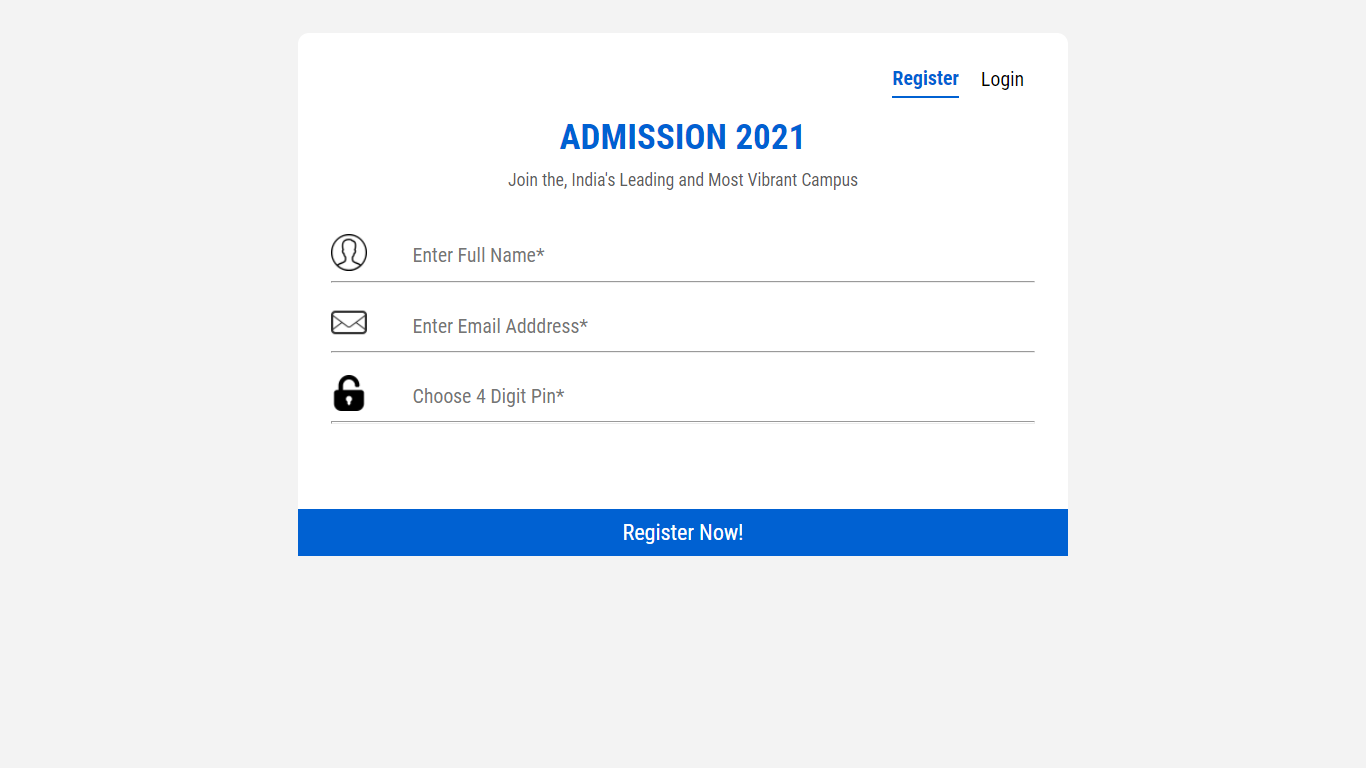
**6.3 Designing Form**

**USER**

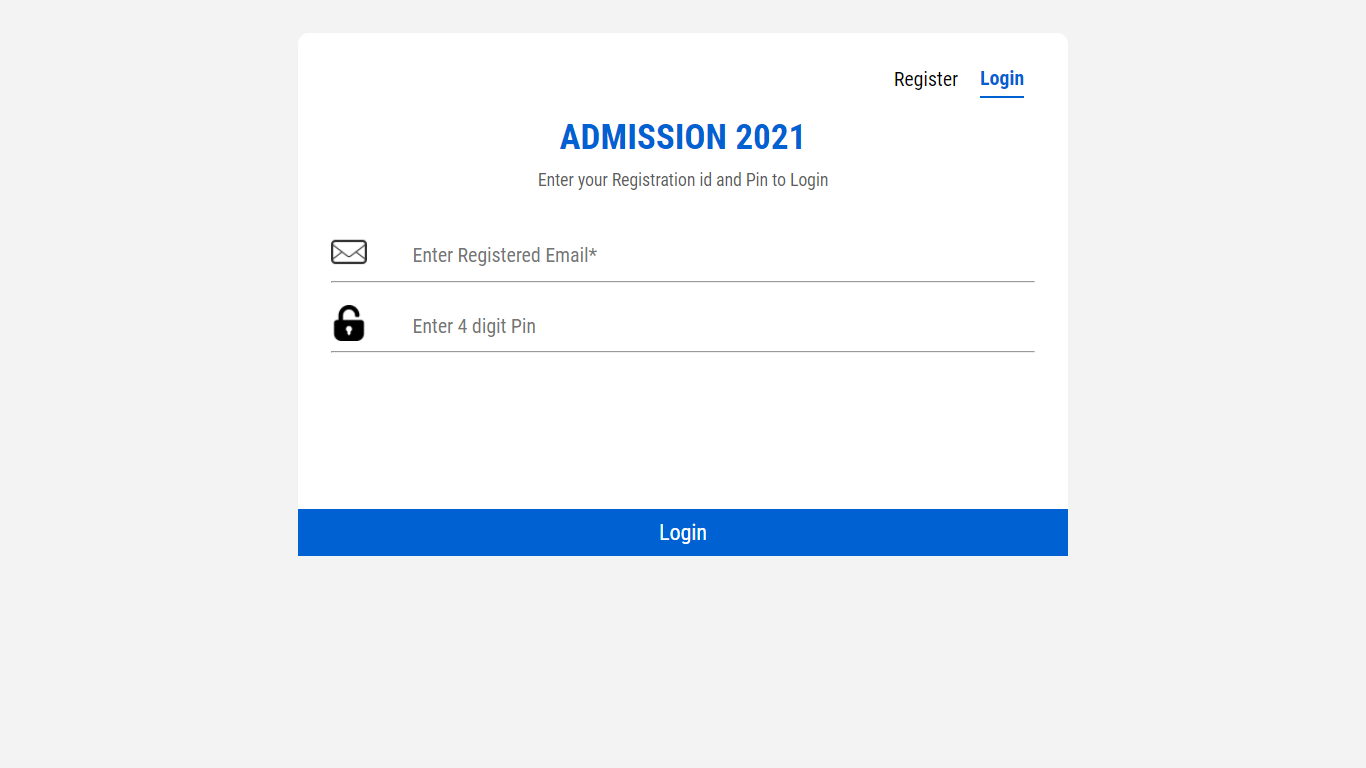
HOME PAGE



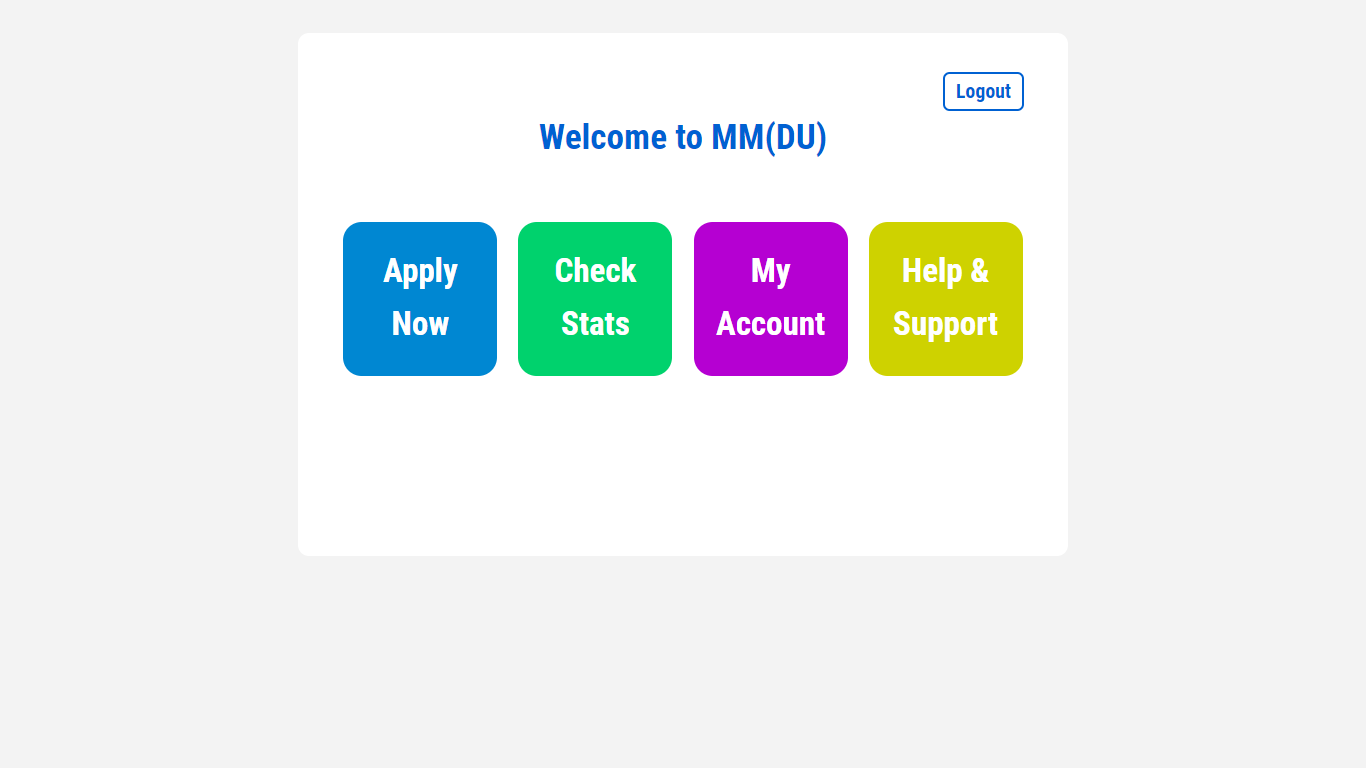
SIGNUP PAGE



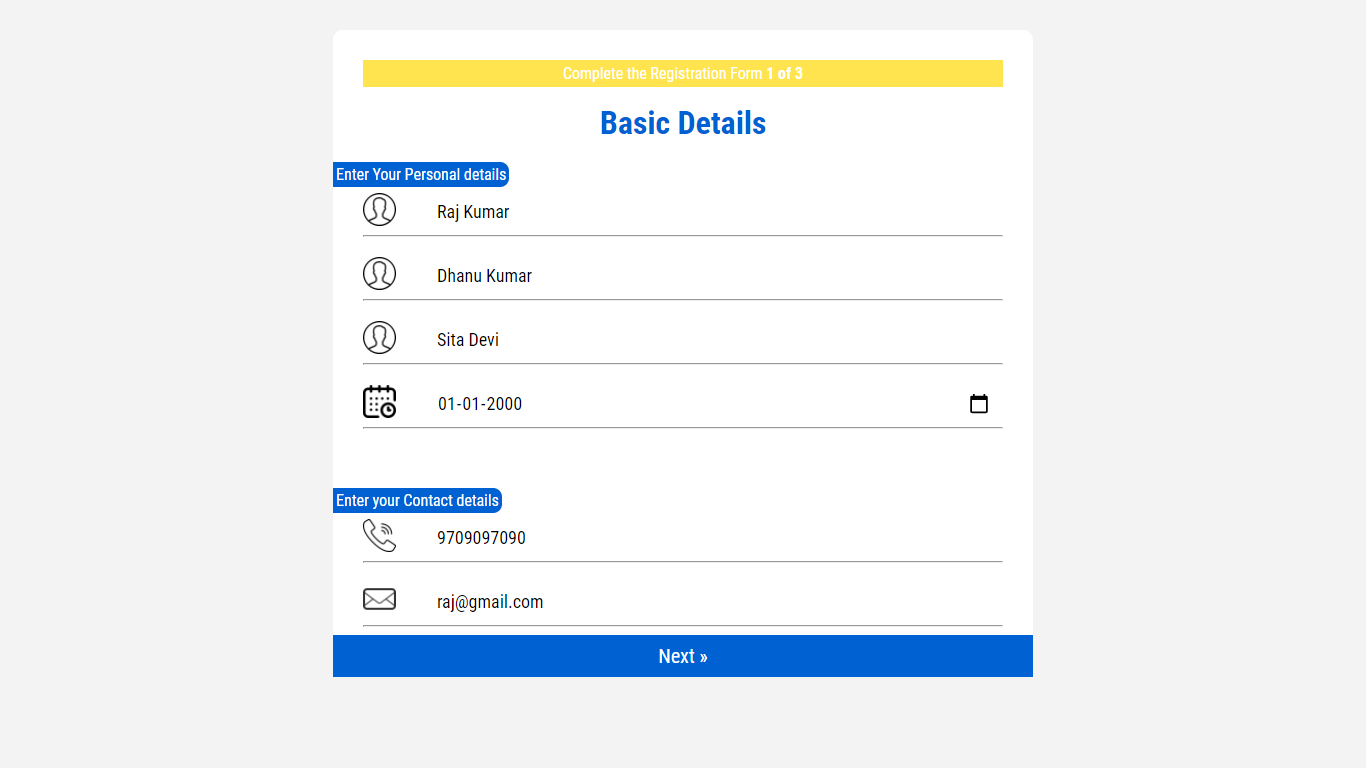
LOGIN PAGE



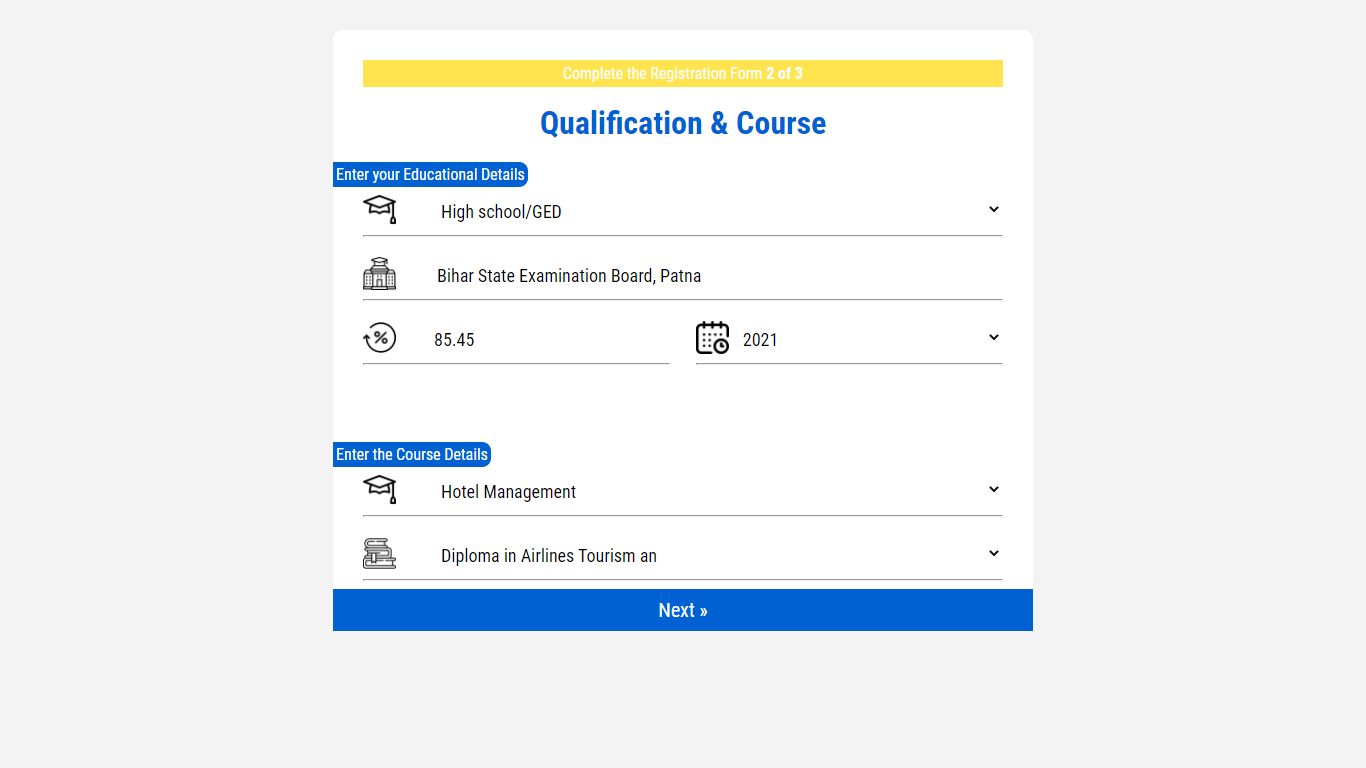
DASHBOARD



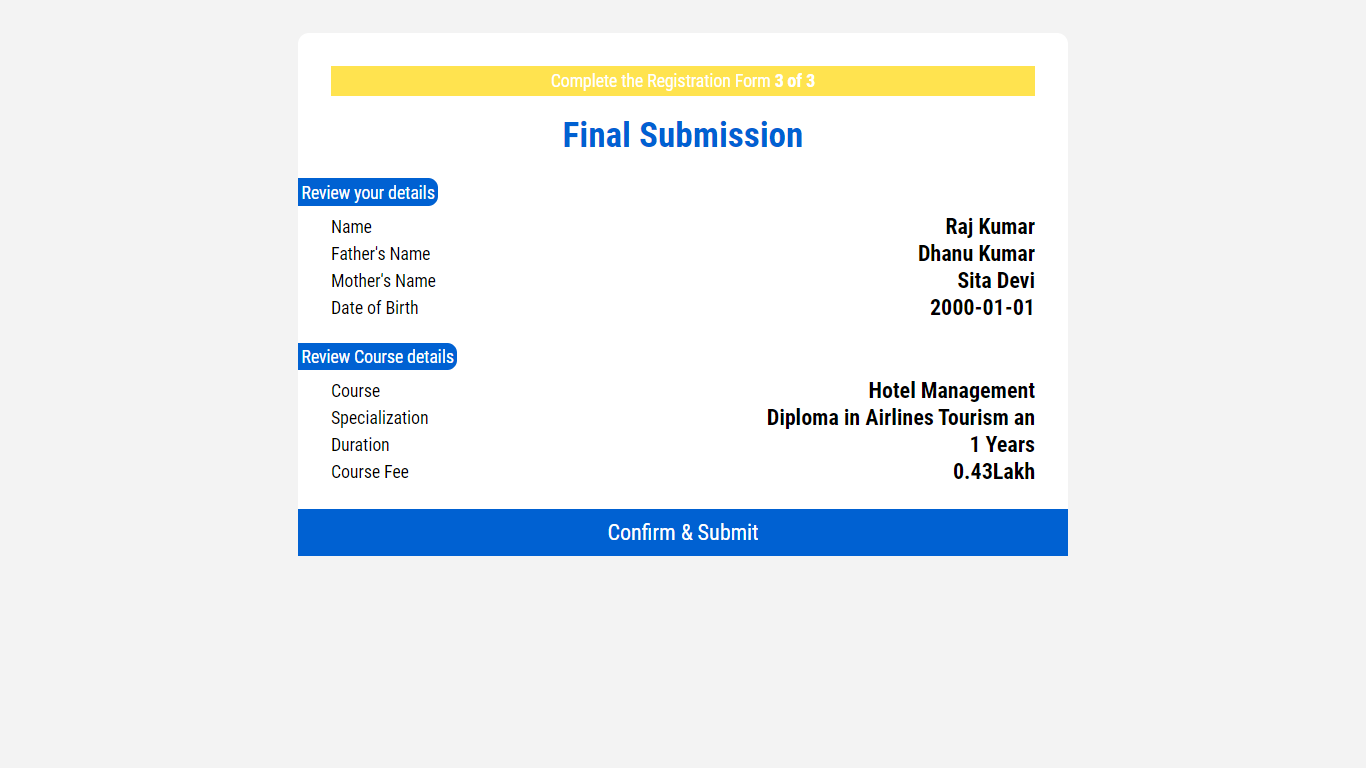
RESISTRATION PAGE 1/3



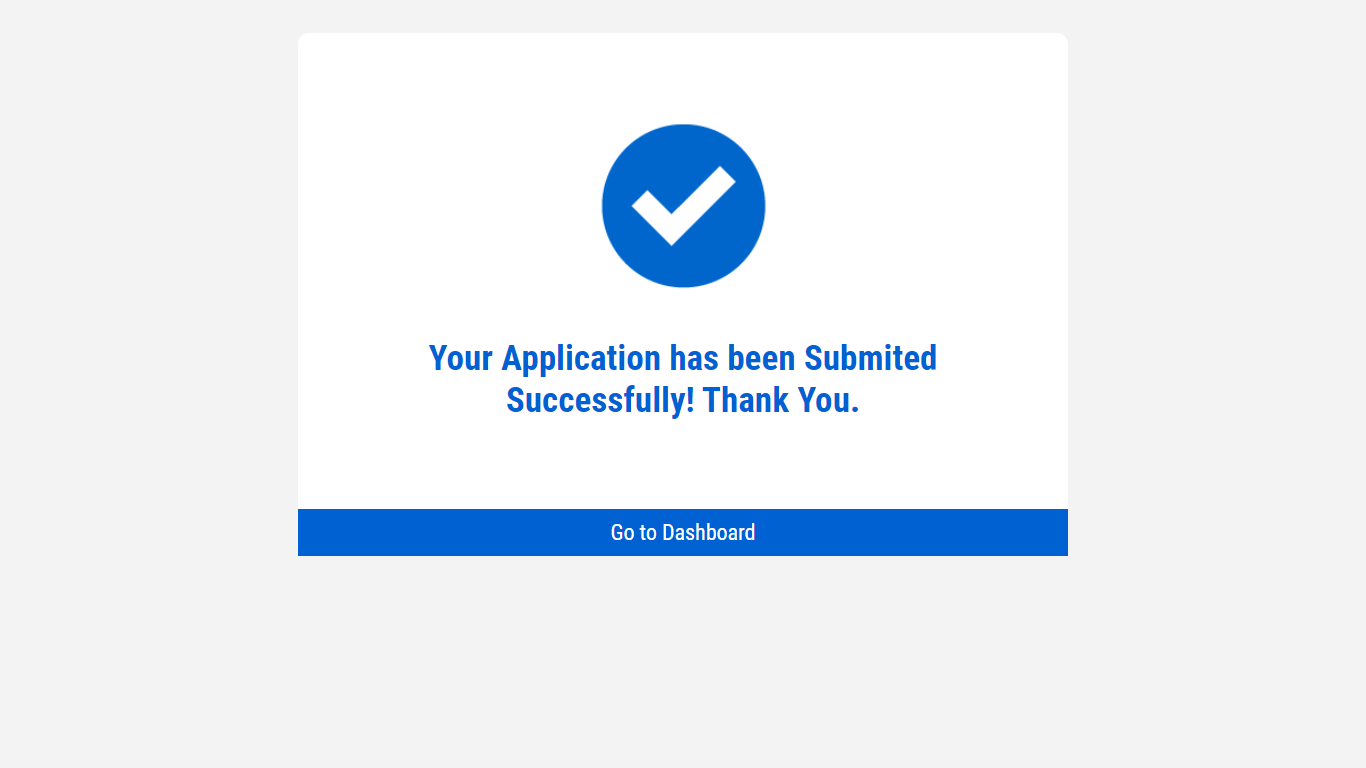
RESISTRATION PAGE 2/3



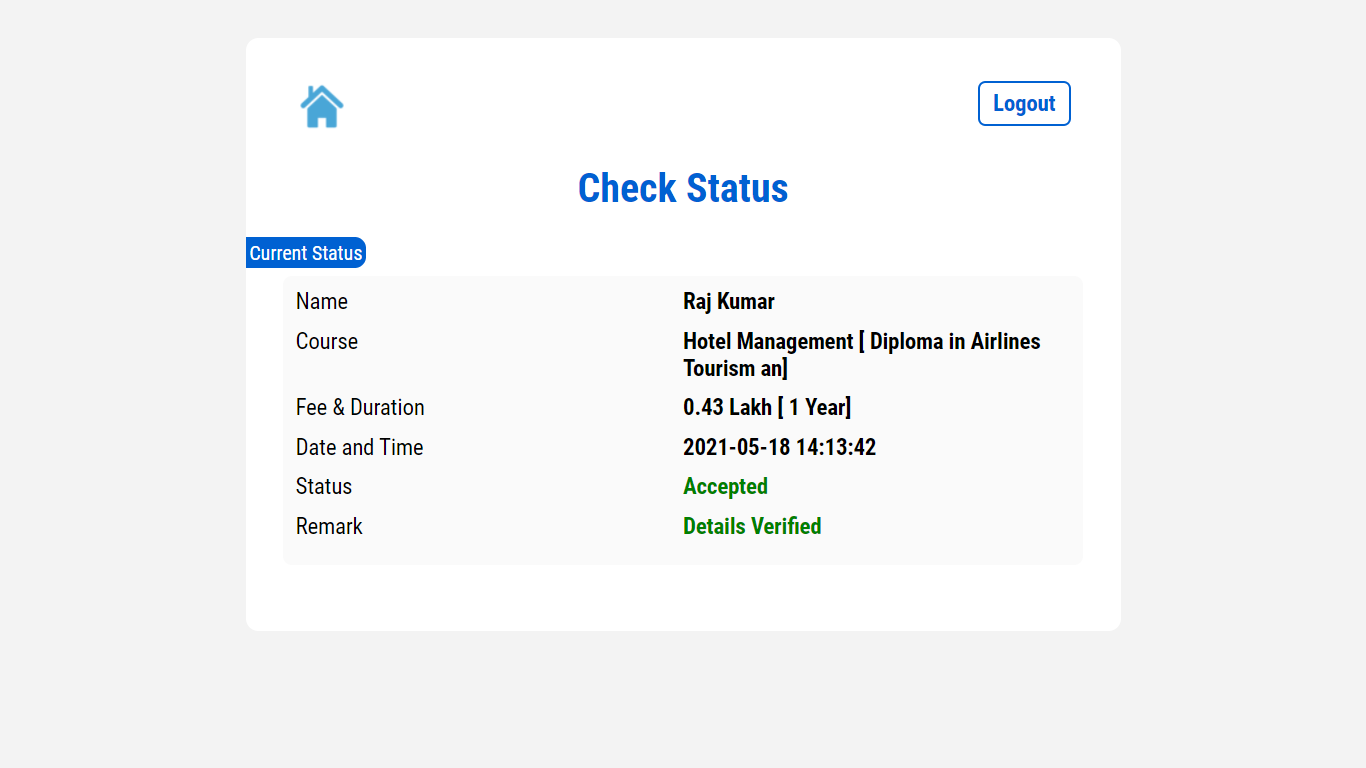
RESISTRATION PAGE 1/3



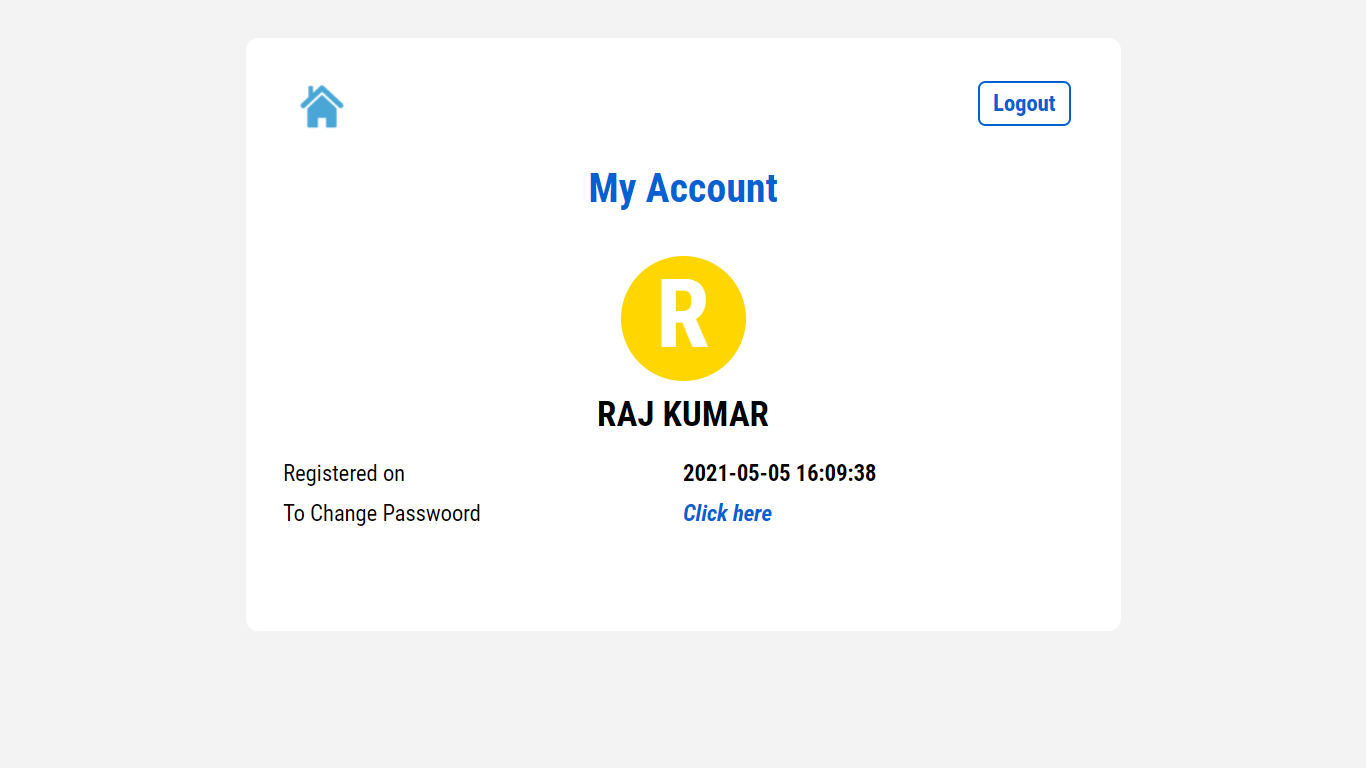
CONFIRMATION PAGE



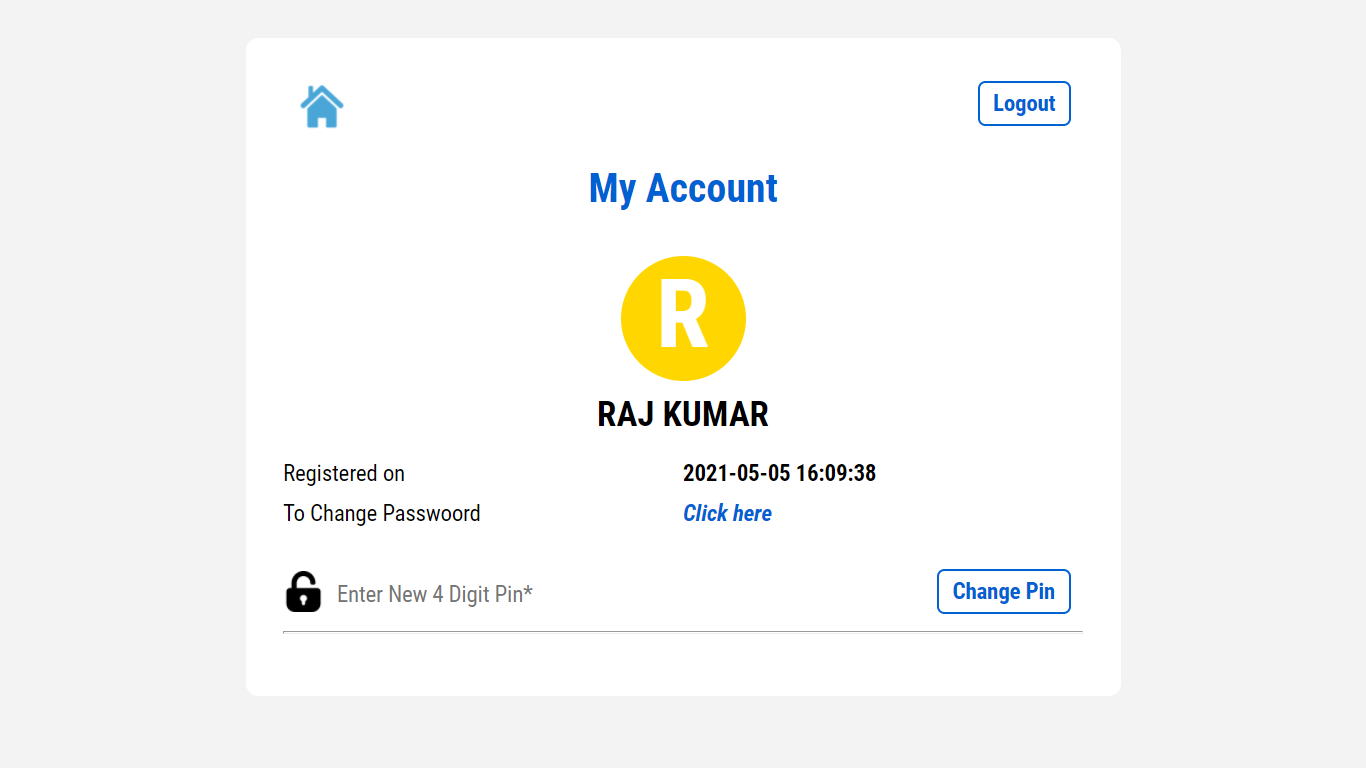
ADMISSION STATUS PAGE



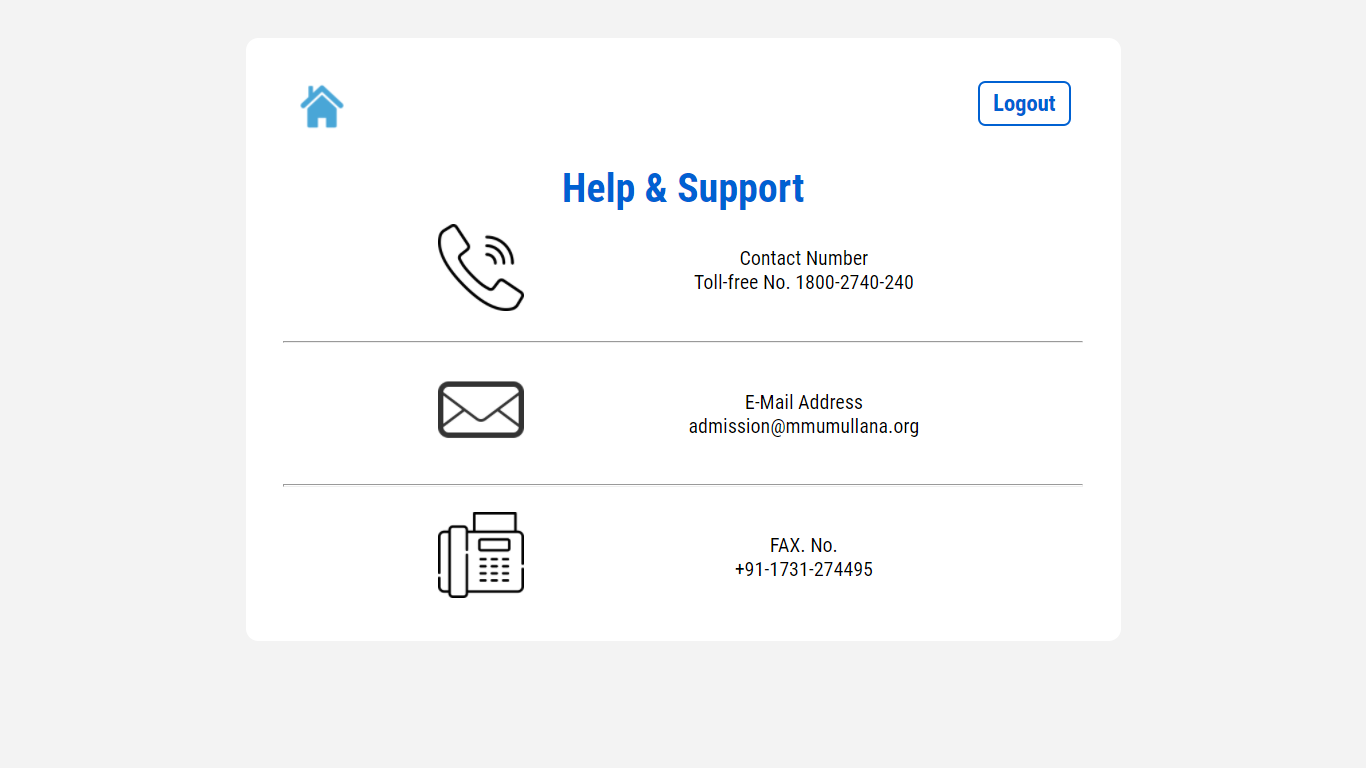
MY ACCOUNT PAGE



PASSWORD CHANGE PAGE

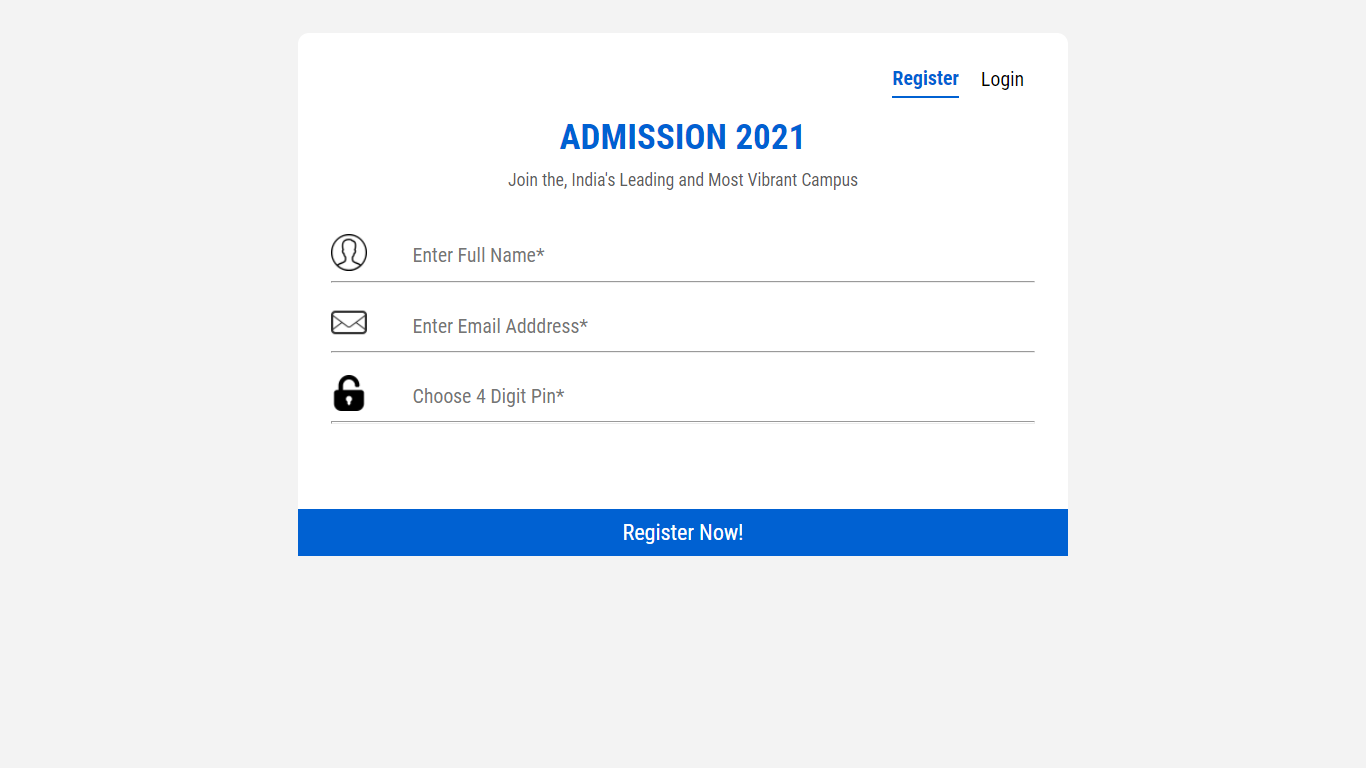


HELP AND SUPPORT PAGE

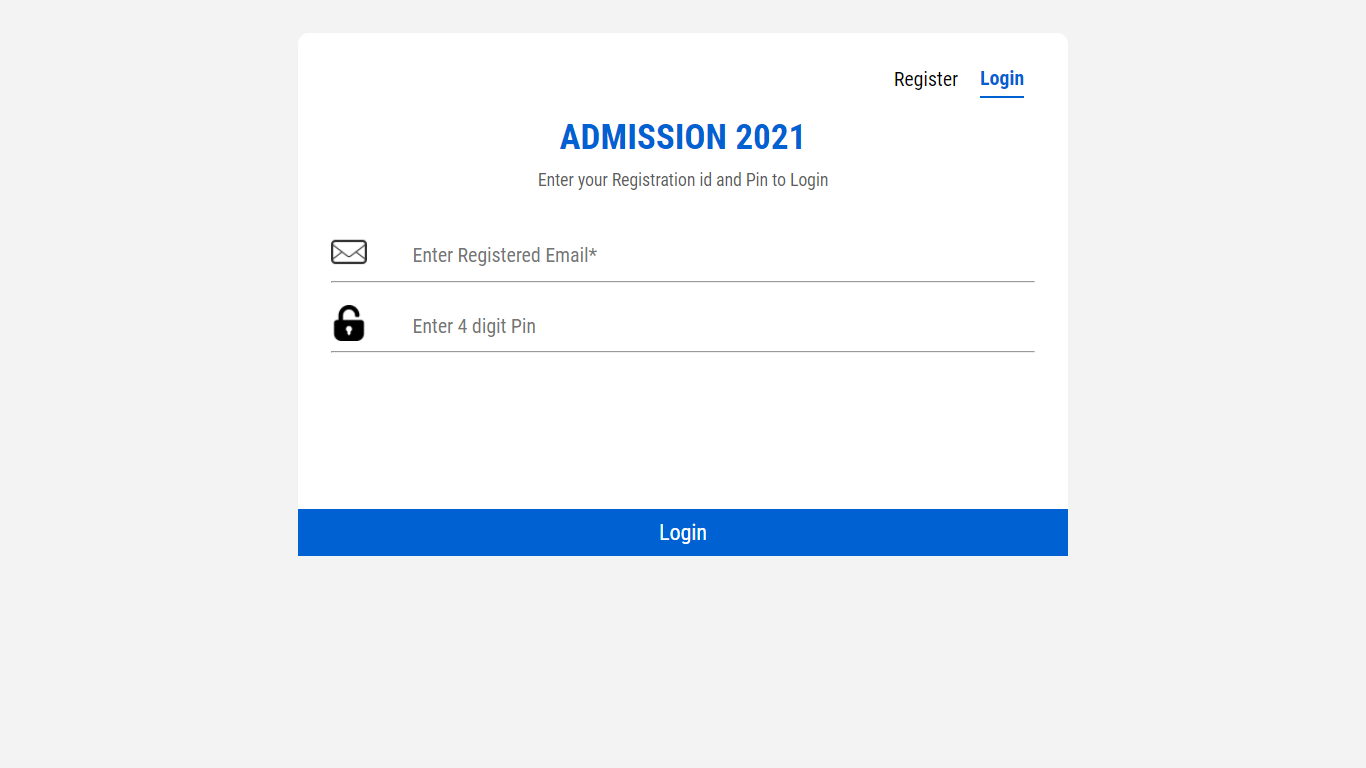


**ADMIN**

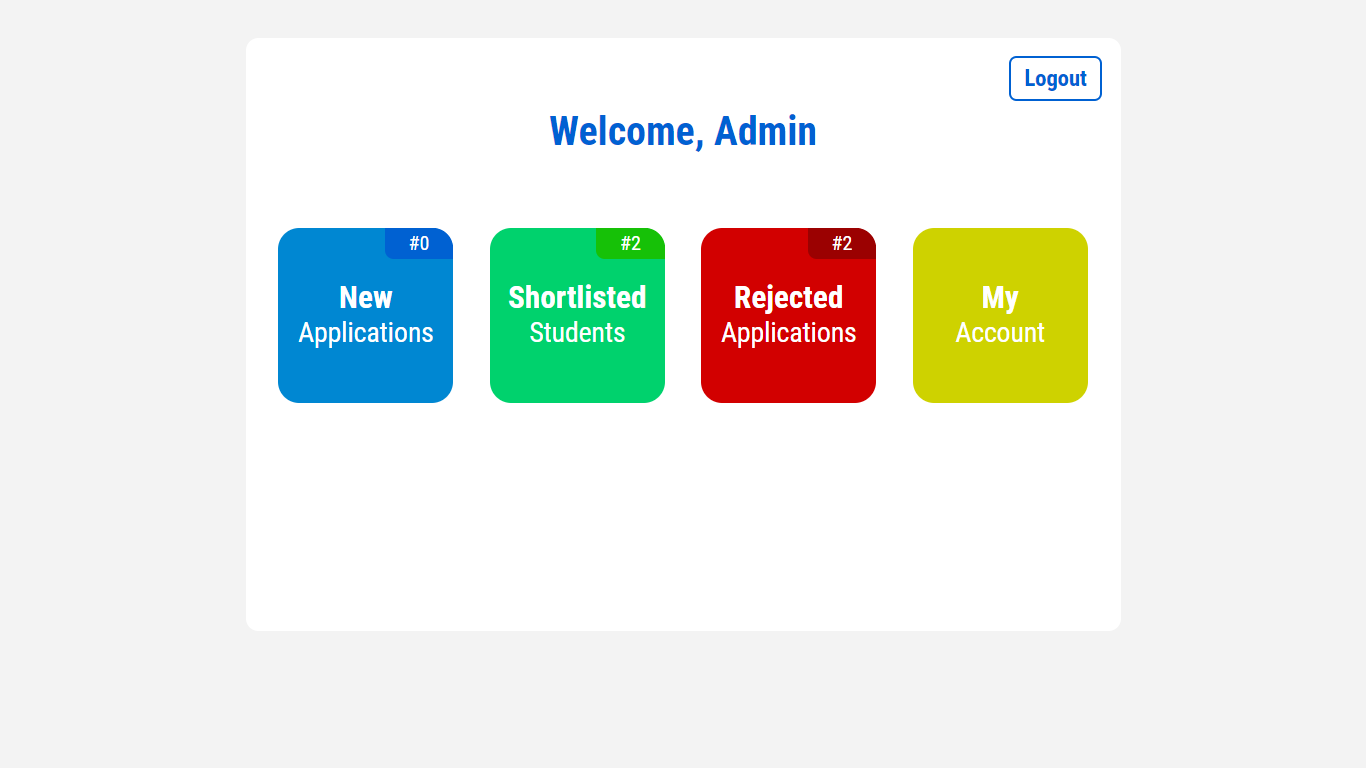
SIGNUP PAGE



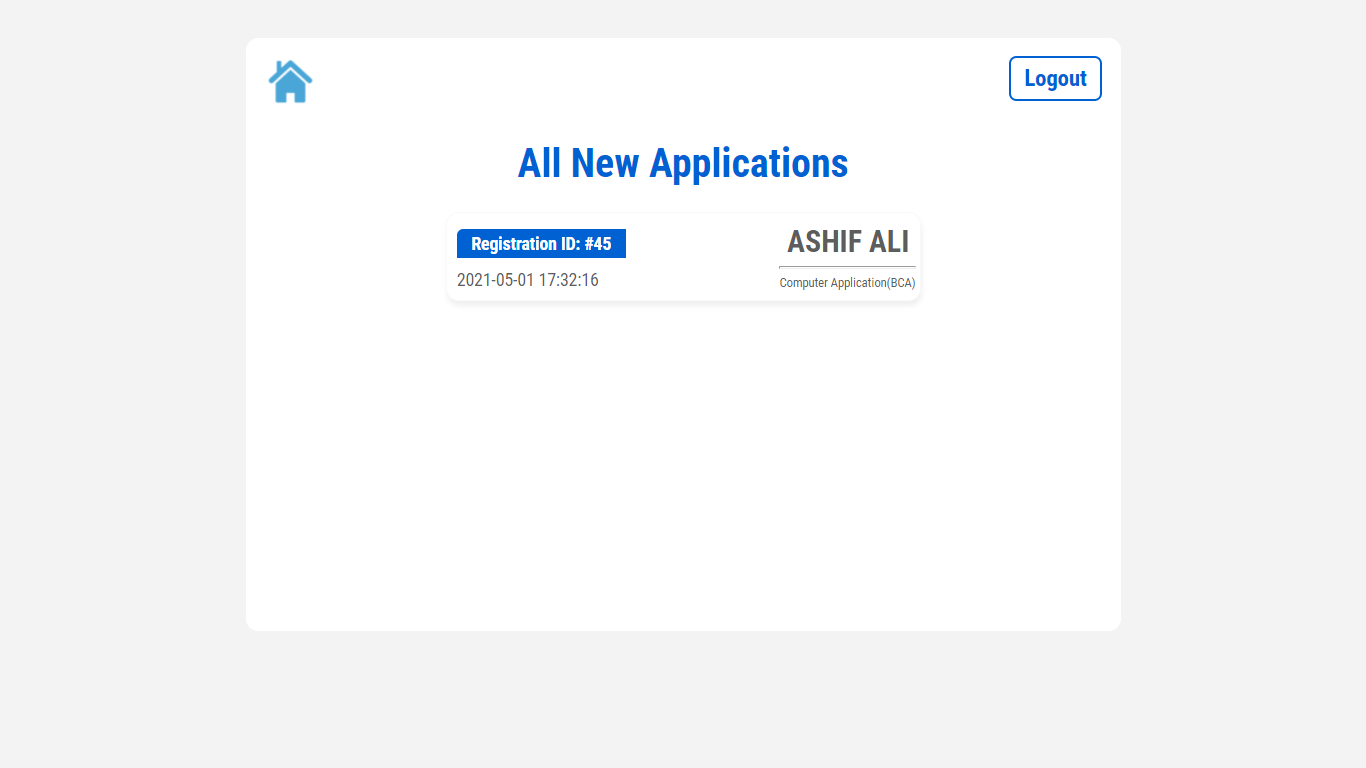
LOGIN PAGE



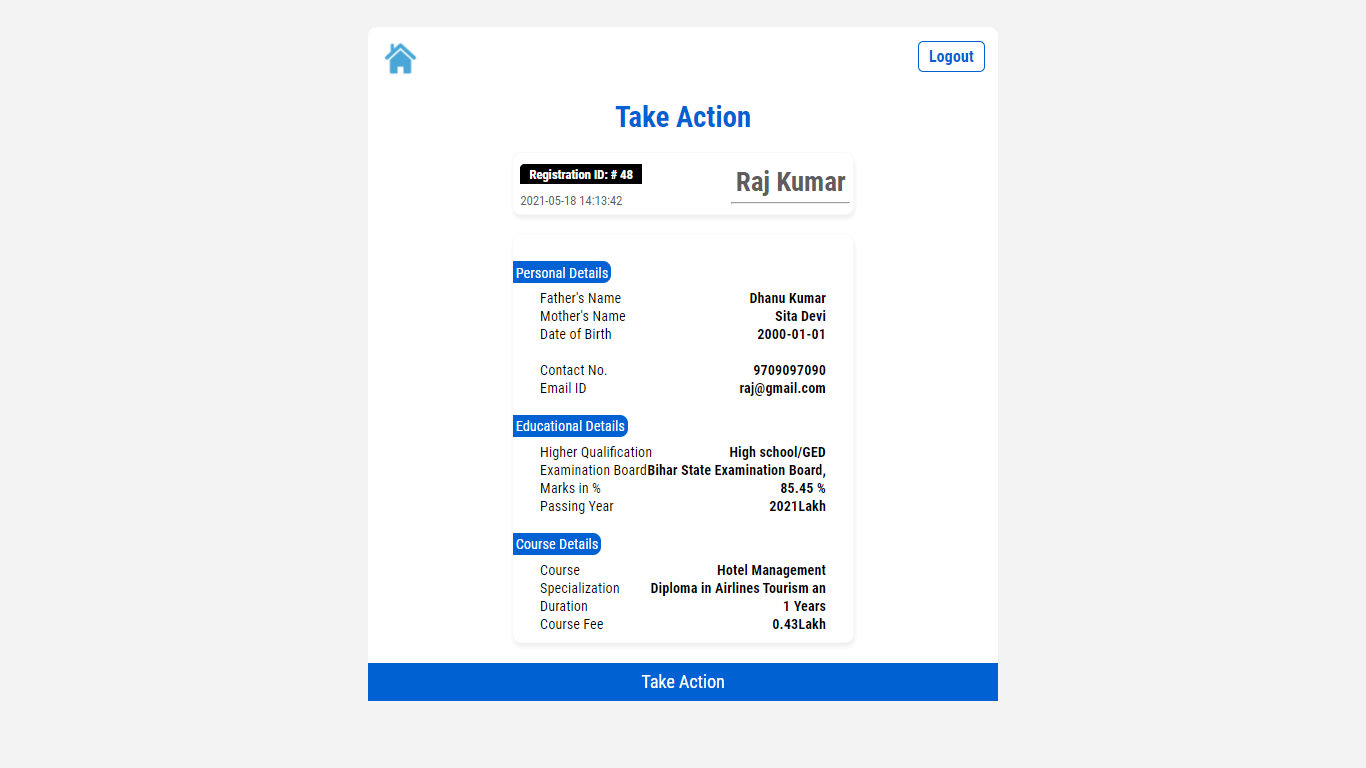
DASHBOARD



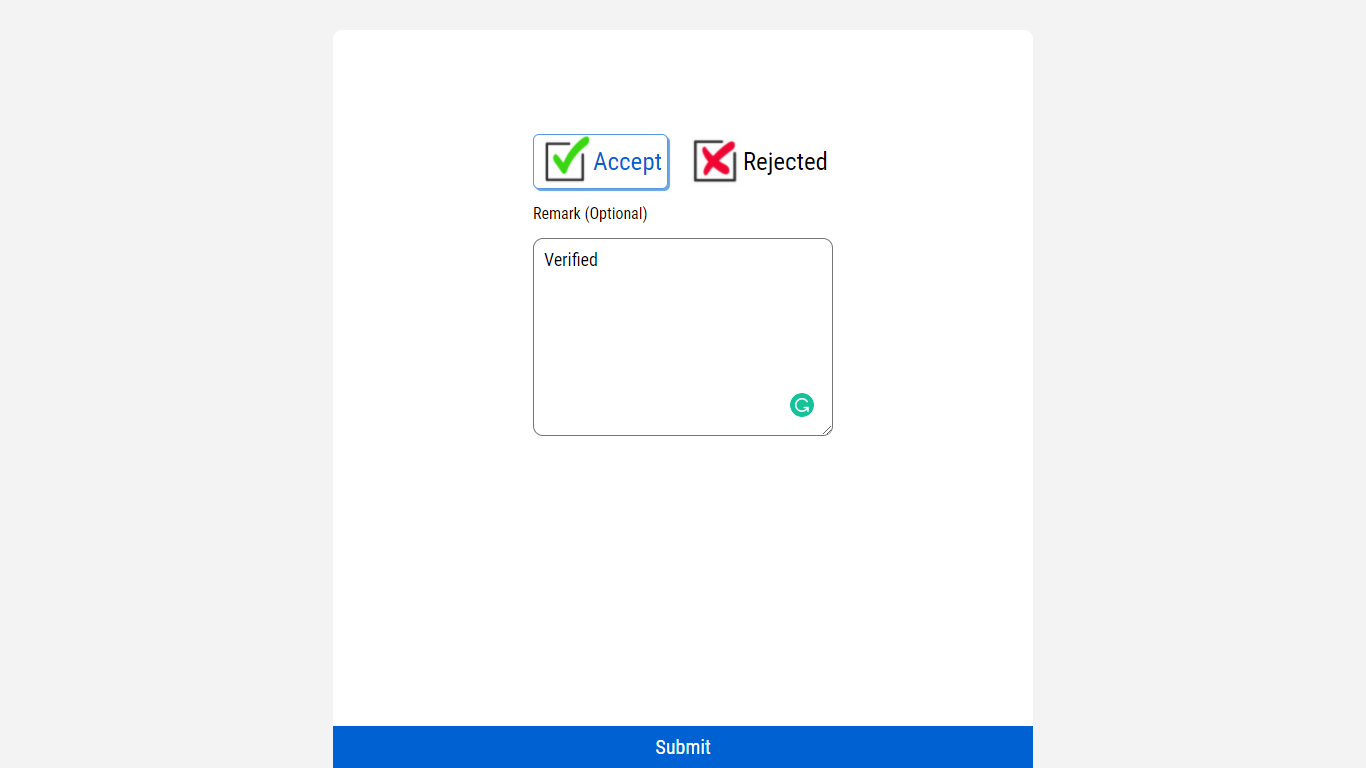
NEW APPLICATIONS PAGE



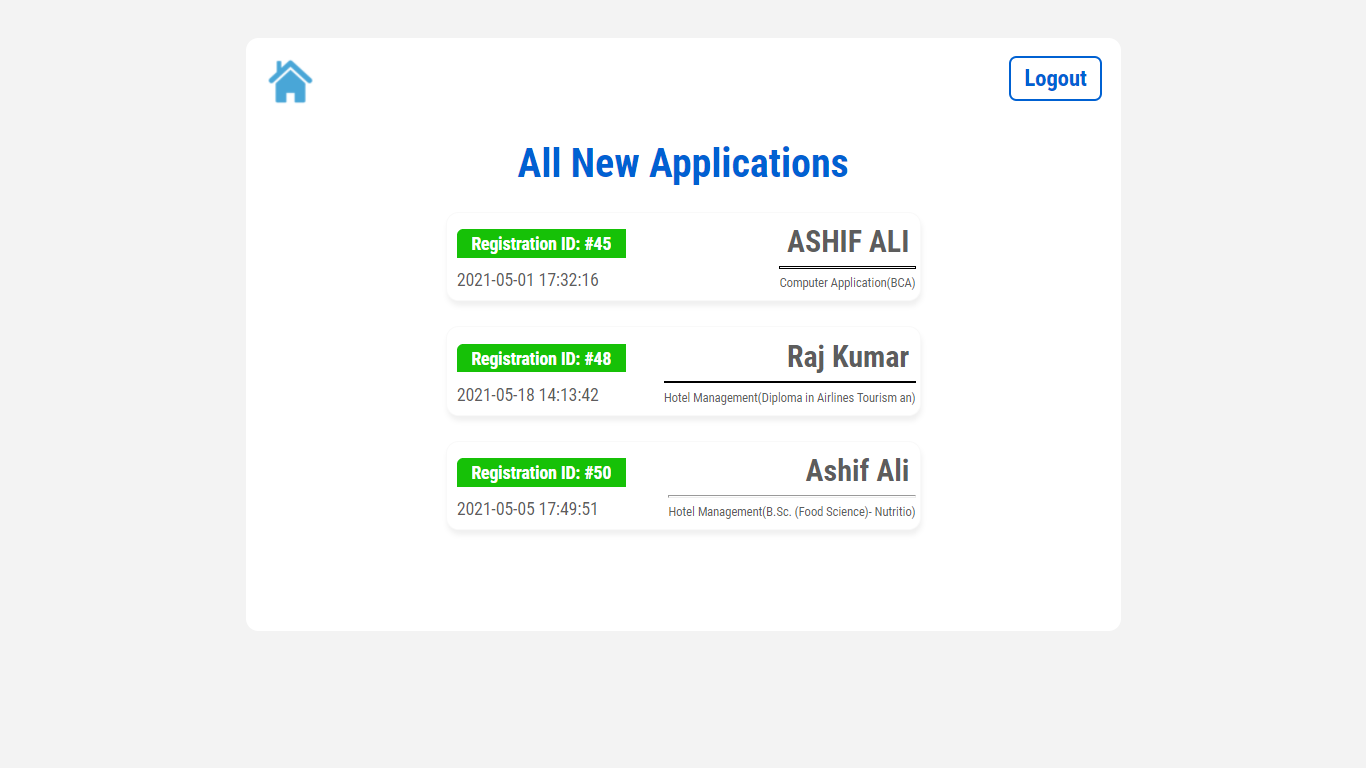
APPLICATION DETAILS PAGE



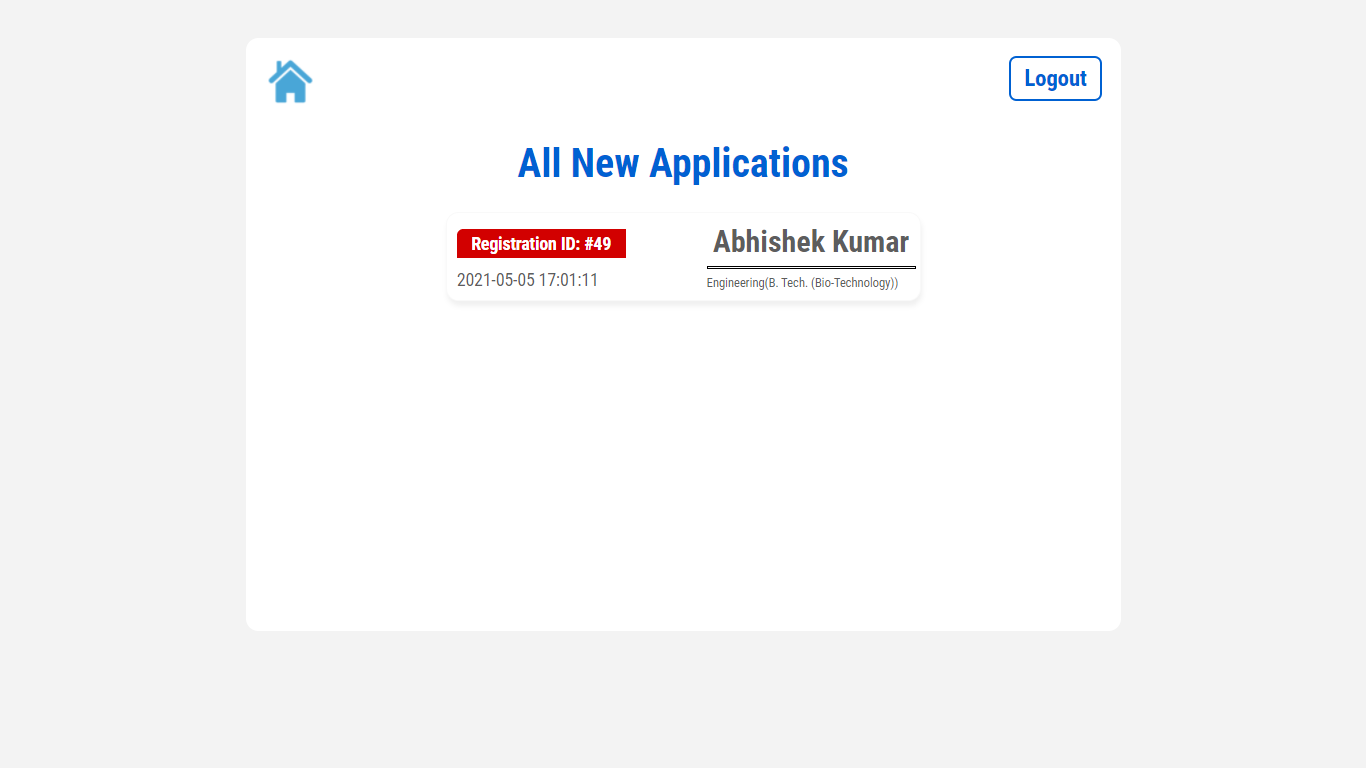
APPLICATION ACCEPT/REJECT ACTION



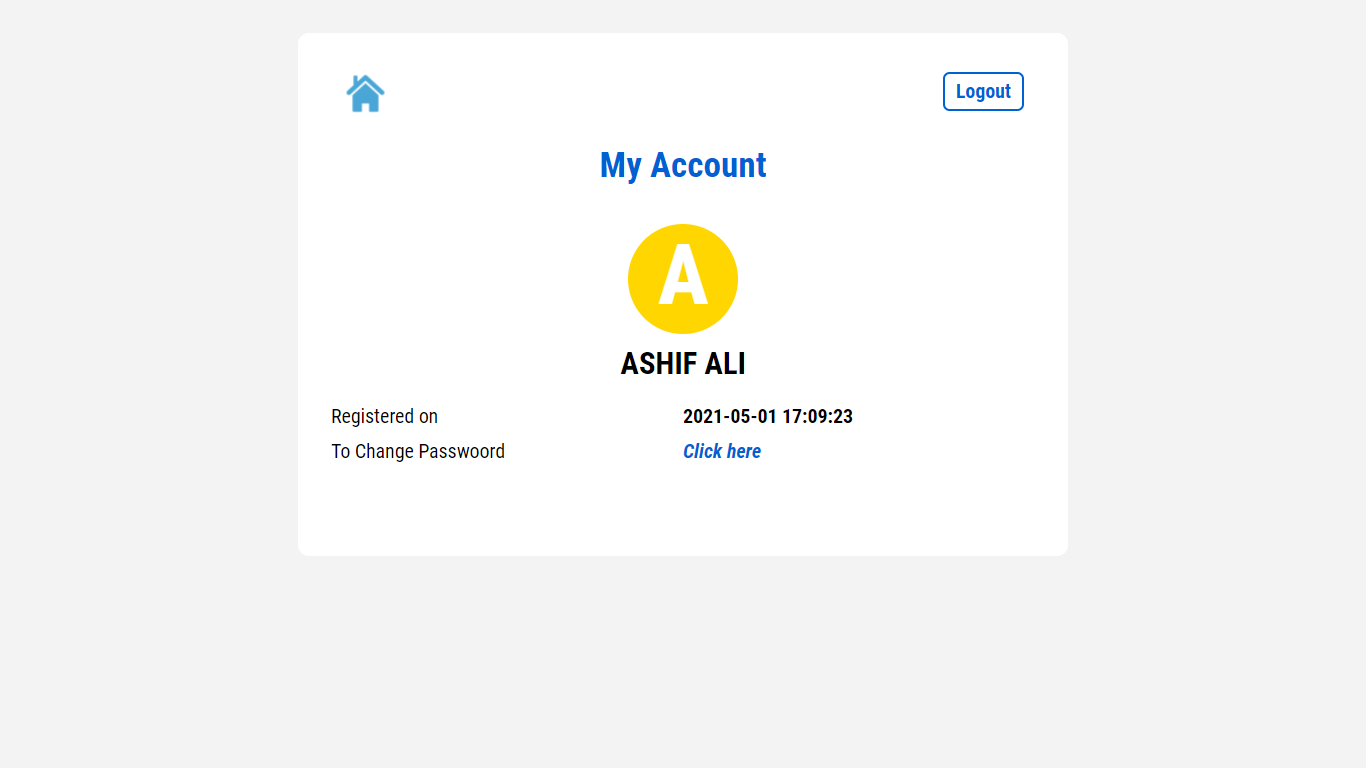
SHORTLISTED APPLICATION PAGE



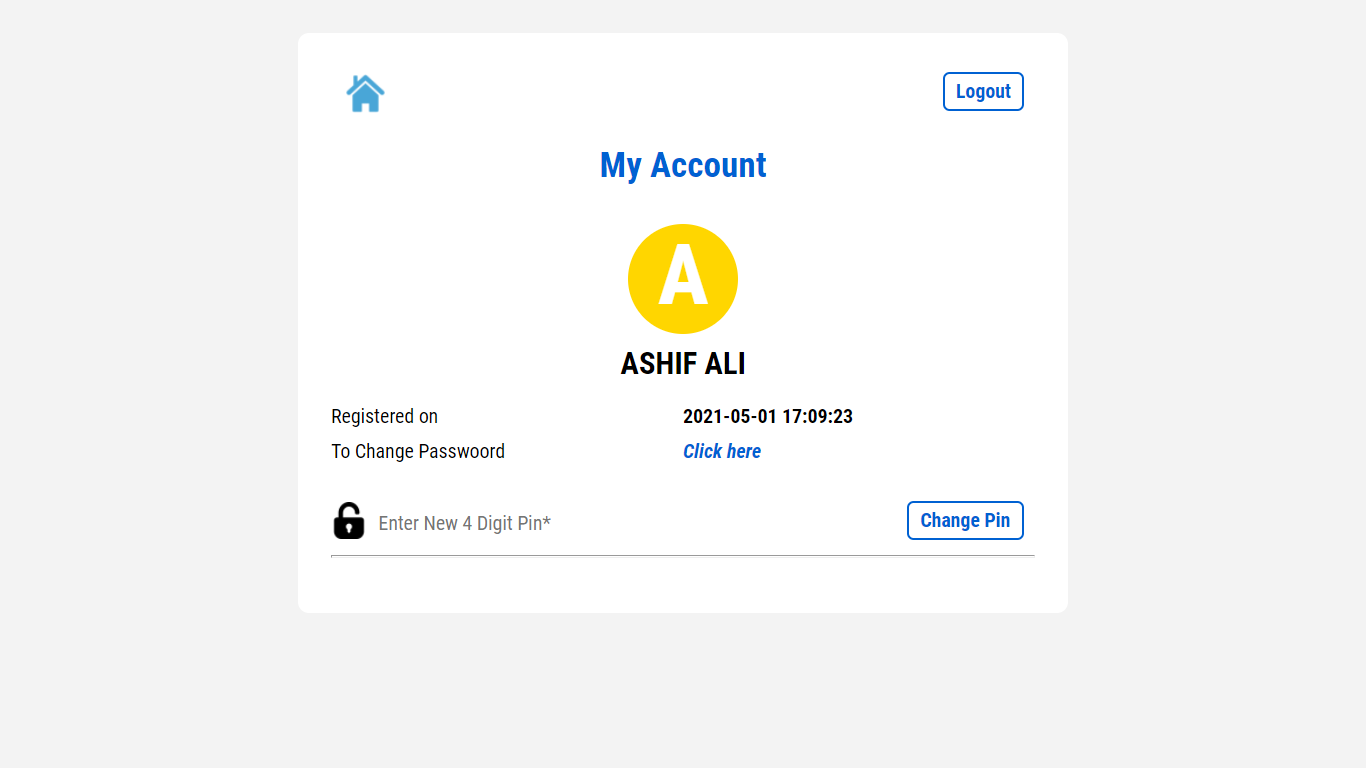
REJECTED APPLICATION



MY ACCOUNT PAGE



CHANGE PASSWORD PAGE



1. **Coding**

**Project category:**

RDBMS (Relational Database Management System)

Project platform:

**Front End:** PHP

**Back End:** MySQL Server

**PHP:**

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. 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. The PHP code is written in the <?php and ?>.

**MySQL:   
MySql**is a powerful database. It's very good and free of charge. Many developers in the world selected mysql and php for developing their website.  MySQL uses a standard form of the well-known SQL data language. MySQL works very quickly and works well even with large data sets.

**Database Coding**

-- phpMyAdmin SQL Dump

-- version 5.1.0

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: May 05, 2021 at 01:18 PM

-- Server version: 10.4.18-MariaDB

-- PHP Version: 7.3.27

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

--

-- Database: `onlineregistration`

--

-- --------------------------------------------------------

--

-- Table structure for table `admin`

--

CREATE TABLE `admin` (

  `admin\_id` int(10) NOT NULL,

  `userType` varchar(10) NOT NULL,

  `admin\_username` varchar(50) NOT NULL,

  `adminName` varchar(30) NOT NULL,

  `admin\_email` varchar(50) NOT NULL,

  `admin\_pin` int(4) NOT NULL,

  `registeredOn` datetime NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- --------------------------------------------------------

--

-- Table structure for table `admission`

--

CREATE TABLE `admission` (

  `admission\_id` int(11) NOT NULL,

  `username` varchar(25) NOT NULL,

  `specializationId` int(10) NOT NULL,

  `shortlisted` varchar(10) NOT NULL DEFAULT 'Pending',

  `remark` varchar(50) NOT NULL,

  `appliedDate` datetime DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Table structure for table `course`

--

CREATE TABLE `course` (

  `courseId` int(11) NOT NULL,

  `courseName` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `course`

--

INSERT INTO `course` (`courseId`, `courseName`) VALUES

(3, 'Agriculture'),

(4, 'Basic Sciences'),

(5, 'Computer Application'),

(6, 'Diploma in Dental'),

(7, 'Engineering'),

(8, 'Hotel Management'),

(9, 'Law'),

(10, 'Management'),

(11, 'Medical/Paramedical Courses'),

(12, 'Nursing'),

(13, 'Pharmacy'),

(14, 'Physiotherapy');

-- --------------------------------------------------------

--

-- Table structure for table `specialization`

--

CREATE TABLE `specialization` (

  `specializationId` int(11) NOT NULL,

  `courseId` int(10) NOT NULL,

  `specializationName` varchar(30) NOT NULL,

  `courseDuration` int(5) NOT NULL,

  `courseFee` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `specialization`

--

INSERT INTO `specialization` (`specializationId`, `courseId`, `specializationName`, `courseDuration`, `courseFee`) VALUES

(1, 5, 'BCA', 3, '2.0'),

(2, 5, 'MCA', 2, '2.9'),

(3, 3, 'B.Sc (Hons) - Agriculture', 2, '4.4'),

(4, 3, 'B.Sc (Hons) - Agriculture Inte', 6, '5.6'),

(5, 4, 'B.Sc. (Hons.) - Bio-Technology', 3, '1.6'),

(6, 4, 'B.Sc Food Science ', 3, '1.6'),

(7, 6, 'Diploma (Dental Hygienist)', 2, '1.15'),

(8, 7, 'B. Tech. (Bio-Technology)', 4, '5.97'),

(9, 7, 'B. Tech. (Computer Science and', 4, '5.97'),

(10, 8, 'B.Sc. (Food Science)- Nutritio', 1, '0.43'),

(11, 8, 'Diploma in Airlines Tourism an', 1, '0.43'),

(14, 9, 'B.Com LLB', 5, '3.45'),

(15, 9, 'LLM', 1, '0.66'),

(16, 10, 'B.Com (Hons.)', 3, '2.32'),

(17, 10, 'MBA', 2, '2.58'),

(18, 11, 'B.Sc. (MLT) (Lateral Entry)', 2, '1.43'),

(19, 11, 'M.Sc. (OTT)', 2, '1.36'),

(20, 12, 'B.Sc. (Nursing)', 4, '3.70'),

(21, 12, 'P.B.B.Sc. (Nursing)', 2, '1.77'),

(22, 13, 'B.Pharma', 4, '4.37'),

(23, 13, 'Pharm-D', 6, '9.67'),

(24, 14, 'MPT', 2, '1.67'),

(25, 14, 'BPT', 4, '3.62');

-- --------------------------------------------------------

--

-- Table structure for table `student`

--

CREATE TABLE `student` (

  `id` int(11) NOT NULL,

  `username` varchar(50) NOT NULL,

  `studentName` varchar(30) NOT NULL,

  `fatherName` varchar(30) DEFAULT NULL,

  `motherName` varchar(30) DEFAULT NULL,

  `dateOfBirth` varchar(20) NOT NULL,

  `mobileNumber` varchar(10) DEFAULT NULL,

  `emailId` varchar(50) NOT NULL,

  `higherQualification` varchar(30) DEFAULT NULL,

  `examinationBoard` varchar(30) DEFAULT NULL,

  `passingPercentage` float DEFAULT NULL,

  `passingYear` int(4) DEFAULT NULL,

  `pin` int(4) NOT NULL,

  `registeredOn` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Indexes for table `admin`

--

ALTER TABLE `admin`

  ADD PRIMARY KEY (`admin\_id`),

  ADD UNIQUE KEY `admin\_username` (`admin\_username`);

--

-- Indexes for table `admission`

--

ALTER TABLE `admission`

  ADD PRIMARY KEY (`admission\_id`),

  ADD UNIQUE KEY `studentId` (`username`),

  ADD KEY `specializationId` (`specializationId`);

--

-- Indexes for table `course`

--

ALTER TABLE `course`

  ADD PRIMARY KEY (`courseId`);

--

-- Indexes for table `specialization`

--

ALTER TABLE `specialization`

  ADD PRIMARY KEY (`specializationId`),

  ADD KEY `courseId` (`courseId`);

--

-- Indexes for table `student`

--

ALTER TABLE `student`

  ADD PRIMARY KEY (`id`),

  ADD UNIQUE KEY `username` (`username`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `admin`

--

ALTER TABLE `admin`

  MODIFY `admin\_id` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;

--

-- AUTO\_INCREMENT for table `admission`

--

ALTER TABLE `admission`

  MODIFY `admission\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=49;

--

-- AUTO\_INCREMENT for table `course`

--

ALTER TABLE `course`

  MODIFY `courseId` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=15;

--

-- AUTO\_INCREMENT for table `specialization`

--

ALTER TABLE `specialization`

  MODIFY `specializationId` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=26;

--

-- AUTO\_INCREMENT for table `student`

--

ALTER TABLE `student`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=36;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `admission`

--

ALTER TABLE `admission`

  ADD CONSTRAINT `admission\_ibfk\_1` FOREIGN KEY (`specializationId`) REFERENCES `specialization` (`specializationId`);

--

-- Constraints for table `specialization`

--

ALTER TABLE `specialization`

  ADD CONSTRAINT `specialization\_ibfk\_1` FOREIGN KEY (`courseId`) REFERENCES `course` (`courseId`);

COMMIT;

1. **Testing**

Different testing levels

* Unit testing
* Integrated testing
* Validation testing
* Output testing
* User acceptance testing

1. **Unit testing:**

Unit testing focuses on verification effort on the smallest unit of software design module. Using the unit test plans. Prepared in the design phase of the system as a guide important control paths are tested to uncover errors within the boundary of the modules. The interfaces of each of the modules under consideration are also tested. Boundary conditions were checked.

All independent paths were exercised to ensure that all statements in the module executed at least once and all error-handling paths were tested. Each unit was thoroughly tested to check if it might fall in any possible situation. This testing was carried out during the programming itself. At the end of this testing phase each unit was found to be working satisfactorily as regarded to the expected out tom the module.

1. **Integration Testing**:

Data can be across an interface one module can have an adverse effect on another's Sub function when combined may not produce the desired major function; global data structures can present problems. Integration testing is a symmetric technique for constructing tests to uncover errors associated with the interface. All modules are combined in this testing step. Then the entire program was tested as a whole.

1. **Validation Testing:**

At the culmination of integration testing software is completely assemble. As a package. Interfacing errors have been uncovered and corrected and find; series of software test-validation testing begins. Validation testing can be defined in many ways but a Simple definition is that validation succeeds when software functions in manner that is reasonably expected by the consumer.

Software validation is achieved through a series of black box tests that demonstrate conformity with requirement after validation test has been conducted one of two conditions exists.

* The function or performance Characteristics confirm to specification that are accepted.
* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

1. **Output testing:**

After performing the validation testing the next step is output testing of the proposed system since a system is useful if it does not produce the required output in the specific format required by them tests the output generator displayed on the system under consideration. Here the output is considered in two ways - one is onscreen and the other is printed format. The output formation the screen is found to be correct as the format was designed in the system design phase according to the user needs. As far as hardcopies are considered it goes in terms with the user requirement Hence output testing does not result any correction in the system.

1. **User acceptance Testing:**

User acceptance of the system is a key factor for success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective System and user at the time of developing and making changes whenever required.

1. **Limitations:**

It cannot be used as offline since it is online program.  
Internet connection is required.  
Basic computer knowledge is required to work on the system.

1. **Future scope and further enhancement of the project**

* In future we can make android mobile based application.
* In future we can add features like web cam based verification system, Certification verification, Interview process through online, Identity card after approval, etc.

1. **Conclusion:**

This website stores admission details submitted by students and college staff. This project will eliminate all the manual intervention and increase the speed of whole process. The system works in Apache server which executes PHP script and MySQL as backend for the database. The system is strong to handle daily operations where the database is cleared over certain time.

1. **Bibliography**

Website: [www.w3schools.com](http://www.w3schools.com)

Website: [www.tutorialspoint.com/php](http://www.tutorialspoint.com/php)

Website: <http://www.tutorialspoint.com/mysql>

Book: Learning PHP MySQL JavaScript and CSS Book by Jason Gilmore