**Alumni Association System**

A project submitted to

**UKA TARSADIA UNIVERSITY**

in partial fulfillment of the requirements for the degree of

**Master of Science**

in

**Information Technology**

for

**5 Years Integrated M.Sc.(IT)**

By

**MANAS PATEL**

**(202204102510002)**

**AND**

**YASH BHAKTA**

**(201906100110046)**

Guided by

**Mr. BHAVIK SARANG**

**ASSISTANT PROFESSOR**



**Babu Madhav Institute of Information Technology**

**Uka Tarsadia University**

**Bardoli – 394350**

**December 2021**

**CERTIFICATE**

This is to certify that **Manas Patel (202204102510002) AND Yash Bhakta(201906100110046)** has submitted project entitled **“Alumni Association System”** as the partial fulfillment for the award of the degree of Master of Science in Information Technology for 5 Years Integrated M.Sc.(IT) in 2022 – 2023.

**Date:**

**Place: BMIIT UTU**

**Mr. Bhavik Sarang Dr. Jitendra Nasariwala**

Guide Programme Coordinator,

**External Examiner**



**Babu Madhav Institute of Information Technology,**

**Uka Tarsadia University,**

**Bardoli – 394350**

**Table of Content**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chapters** | **Particulars** | | | **Page no.** |
| **1** | **Introduction** | | |  |
| 1.1 | Problem Definition | |  |
| 1.2 | Initial Requirement Document | |  |
| 1.3 | Project Objective | |  |
| 1.4 | Product Scope | |  |
|  | | | | |
| **2** | **Overall Description** | | |  |
| 2.1 | Product Perspective/ Environment Description | |  |
|  | 2.1.1 | Hardware Interface/ Hardware Specification |  |
|  | 2.1.2 | Software Interface/ Software Specification |  |
|  | | | | |
| **3** | **System Planning** | | |  |
| 3.1 | Software Engineering Model | |  |
|  | 3.2 | CPM PERT | |  |
|  | | | | |
| **4** | **System Specific Requirements** | | |  |
| 4.1 | Functional Requirement | |  |
| 4.2 | Non-functional Requirement | |  |
|  | | | | |
| **5** | **System Analysis** | | |  |
| 5.1 | Use case Diagram | |  |
| 5.2 | Activity Diagrams | |  |
|  | | | | |
| **6** | **System Design** | | |  |
| 6.1 | Database Design | |  |
| 6.1.1 | Database Schema |  |
| 6.1.2 | Data Dictionary |  |
|  | | | | |
| **7** | **System Implementation** | | |  |
| 7.1 | Screenshots | |  |
|  | | | | |
| **8** | **Testing** | | |  |
| 8.1 | Test Cases | |  |
|  | | | | |
| **9** | **Future Enhancement** | | |  |
|  |  | | |  |
|  | **Conclusion** | | |  |
|  |  | | |  |
|  | **Bibliography** | | |  |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Figure Description** | **Page no.** |
| **5.1.1** | **Use Case Diagram for Admin** |  |
| **5.1.2** |  |  |
| **5.1.3** |  |  |
| **5.2.1** | **Activity Diagram for Manage Product** |  |
| **5.2.2** |  |  |
| **5.2.3** |  |  |
| **5.2.4** |  |  |
| **5.2.5** |  |  |
| **5.3.1** | **Sequence Diagram for “adding a product”** |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Table Description** | **Page no.** |
|  | Data Dictionary Tables |  |
| 8.1.1 | Test Cases for Login |  |
| 8.1.2 | Test Cases for Product |  |

**1. INTRODUCTION**

**1.1 PROBLEM DEFINITION :**

Every super-store needs to maintain the stock, employee information, customer's information which leads to so much paperwork where there are chances of losing some or the other data. Online Grocery store Management System helps the super-store to maintain all this information at one place without losing the data.

At present the super-store have lots of items which is hard to control also if there are so many customers it takes long time for payment. In case of out of stock for some items the owner loses the customer some of its earning. It is also hard to maintain all the records of stock, employees and customer. The sales become limited for that particular area which impacts on the income of the owner.

In Online grocery store management system, you get updated whenever you are going out of stock for any items. There is no rush as like in the shop so it becomes easy to choose your items as well as for doing the payment. All the records of stocks, sales, employee and customer can be easily stored and found easily whenever needed. Online sailing increases the profit as the reachability of the customer increases.

**1.2 INITIAL REQUIREMENT DOCUMENT:**

|  |  |
| --- | --- |
| Title of the project | Alumni Association System |
| Stakeholders involved in capturing requirements | Project Guide, Institute Coordinator, Alumni’s |
| Techniques used for requirement capturing | Brainstorming, Interview |
| Name of the person along with designation | Mr. BHAVIK SARANG(Assistant Professor)  YASH BHAKTA(Student)  MANAS PATEL(Student) |
| Date | August ‘2022 |
| Users of the system | Administrator, Institute Coordinator, Alumni Students |
| Version | 1.0 |
| Consolidated list of initial requirements:   1. The system shall be able to authenticate the administrator to use the system. 2. The administrator shall be able to manage institute. 3. The administrator shall be able to manage department (institute wise). 4. The administrator shall be able to manage academic years. 5. The administrator shall be able to manage institute coordinator. 6. The administrator shall be able to manage students (who will be an alumni) department wise. 7. The system shall be able to generate login ID and password to the system user that is the institute coordinator and the alumni students. 8. The institute coordinator and the alumni students shall be able to first update their profile and change password. 9. The institute coordinator and alumni students shall be able to manage posts which they have been posted; like information about job vacancies, seminars, workshops, STTPs or any other information related to their courses 10. Both the institute coordinator and alumni students shall be able to view the post vice versa. 11. The institute coordinator and alumni students shall be able to view the profile vice versa, so that they shall be able to communicate with each other through email or calls. 12. The institute coordinator and alumni student(s) shall also be able to ask queries to each other in the form of request or in the form of personal posts like chat box; and they shall be able to view and give response to the queries. 13. The system should be able to generate reports like:     1. Details of the department     2. Details of all institute coordinators:        * Year Wise        * Department Wise     3. Details of all alumni students:        * Year Wise        * Department Wise | |

**1.3 PROJECT OBJECTIVE:**

* To reduce paper work.
* To provide an easy communication between institute and student.
* Save time and efforts for maintaining student details.
* To bring transparency and fairness.

**1.4 PRODUCT SCOPE:**

* The scope of the system is limited for specific institute.
* The scope includes all semester for specific acedmic year.

**2. Overall Description**

**2.1 Product Perspective/Environment Description**

**2.1.1 Hardware Interface/Hardware Specification**

The following is a list of minimum requirements on server side.

Processor : i5 dual core or more

Speed : 2GHz

Memory : 2GB

Hard Disk : 100 GB

Interface : Mouse, Keyboard

On client side any hardware that can run web browser.

**2.1.2 Software Interface/Software Specification**

**Software Requirements**

This web site requires following software in Server and client device

**Server-side Requirements**

Operation System : Windows 7 or higher

Web Server : Local host

Framework : React js, Node js

Database : Mongo DB

Minimum Space : 10 GB

**Client-side Requirements**

Browser : All Browser

**3. System Planning**

**3.1 Software Engineering Model**

**4. System Specific Requirements**

**4.1 Functional Requirements:**

4.1.1 Manage User Login/Registration

4.1.2 Manage Students

4.1.3 Manage Institute Coordinator

4.1.4 Manage Sub Categories

4.1.5 Manage Product

4.1.6 Manage Customer

4.1.7 Manage Customer Order

4.1.8 Manage Cart

4.1.9 Manage Feedback

4.1.10 Manage Payment

4.1.11 Generate Report

4.1.1 Manage User Login

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| FR1 | The module will be used by Administrator, Institute Coordinator and Alumni Students. | Manage User Login |
| FR2 | The Administrator, Institute Coordinator, Alumni Students will be allowed to login in the system using Username and Password. | User\_Login |
| FR3 | The system will authenticate Username and Password. | User\_Login |
| FR4 | User can change or reset the password | Change\_Password, Forgot\_Password |

4.1.2 Manage Student

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| FR1 | The administrator and institute cordinator will be able to manage this module. | Manage\_Student |
| FR2 | The administrator will be able to store details of student like Name, Email, Password, Contact, Address. | Add\_Student |
| FR3 | Administrator and Institute Coordinator can update the details. | Update\_Student |
| FR4 | Administrator and Institute Coordinator can view all the details of Student and Student can view his/her details. | View\_Student |

4.1.3 Manage Institute Coordinator

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| FR1 | The administrator will be able to manage this module. | Manage Institute Coordinator |
| FR2 | The administrator will be able to add institute coordinator details | Add\_Institute\_coordinator |
| FR3 | The administrator will be able to allocate institute coordinator to institute | Allocate\_institute |
| FR4 | The administrator can view all the details of Institute Coordinator and Institute Coordinator can view his/her details. | View\_institute\_coordinator |
| FR5 |  |  |
|  |  |  |
|  |  |  |

4.1.4 Manage Sub Categories

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| FR1 | The admin will be able to manage this module. | Manage\_Product |
| FR2 | Admin can add any new Categories. | Add\_SubCategories |
| FR3 | Admin can active/deactive categories and can also search and view categories. | View\_SubCategories  Search\_SubCategories  Remove\_SubCategories |

4.1.5 Manage Products

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| FR1 | The admin will manage this module | Manage\_Product |
| FR2 | Unique id will be assign to each product by the system. | Add\_Product |
| FR3 | The admin will add the product and will the details of the product like Product name, its category, price, brand, quantity, etc | Add\_Product |
| FR4 | The admin will be able to add new product, update an existing product. | Add\_Product |
| FR5 | Admin will be able to activate/deactivate product also admin can search the product from product category, price, brand | View\_Product  Search\_Product  Remove\_Product |

4.1.6 Manage Customer

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| FR1 | The admin, staff and customer will manage this module. | Manage\_Customer |
| FR2 | Customer can get registered in the system. Customer needs to provide the details like customer name, email id, mobile no, password, address, city, pincode for registration. | Customer\_Registration |
| FR3 | The Customer can update their details. | Update\_Customer |
| FR4 | Admin and staff can view and search customers. | View\_Customer  Search\_Customer |

4.1.7 Manage Customer Order

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| FR1 | The module will be managed by Admin, staff and customer. | Manage\_Customer\_Order |
| FR2 | The Customer can manage their product cart. Customer can add product in cart, update cart, remove product from cart. | Add\_To\_Cart  Update\_Cart  Remove\_From\_Cart |
| FR3 | Customer will be able to place the order after selecting the payment mode. | Place\_Order |
| FR4 | Customer can view order placed and order history.  Also customer can cancel the order. | My\_Order  Cancel\_Order |
| FR5 | Admin and staff can view customer’s order that is placed order, canceled order, delivered order. | View\_Order |

**4.2 Non-Functional Requirement**

1. Security

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| NFR1 | The application is password protected. The user must enter correct username and password to enter in the system | Login Form |

2. Availability

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| NFR1 | The application is available for all the time. | - |

3. Maintainability

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| NFR1 | The system is able to maintain all the details of the institute event. | Admin |

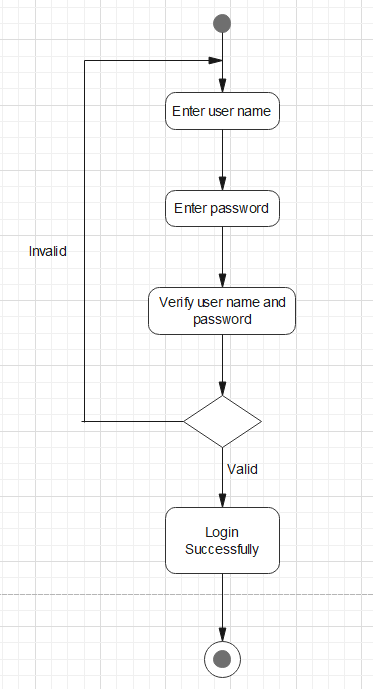
**5. System Analysis**

**5.1 Use Case Diagram**

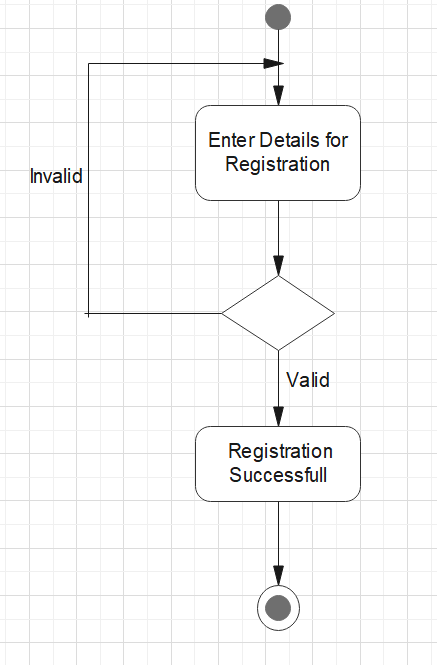


**5.2 Activity Diagram**

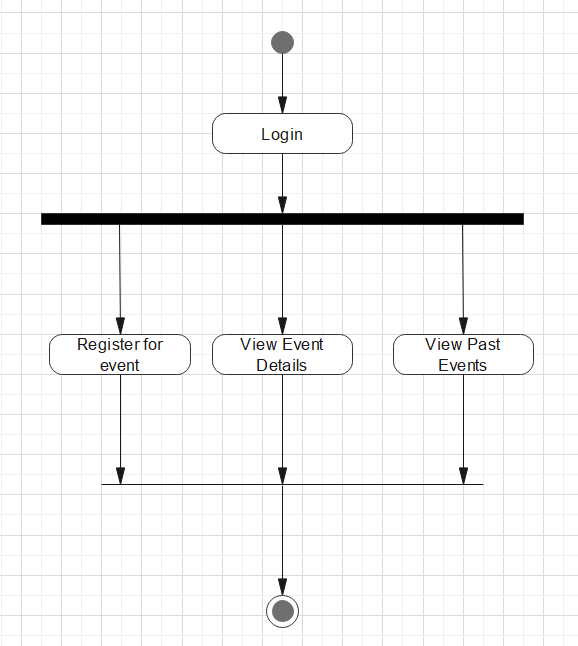
**-Activity Diagram for login**



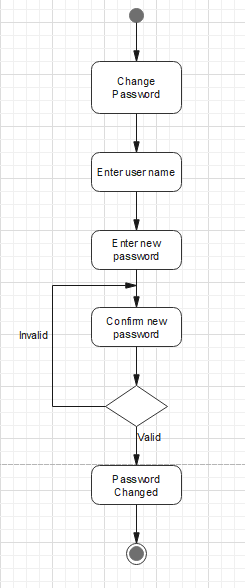
**-Activity Diagram for Registration of Staff and Student**



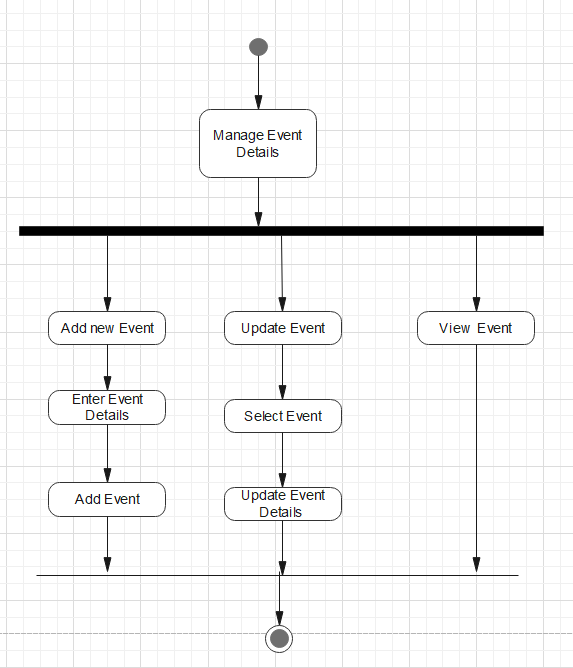
**-Activity Diagram for Registration for the event by student**



**-Activity Diagram for Change Password**



**-Activity Diagram for Manage Event:**



**6. System Design**

**6.1 Database Design**

**6.1.1 Database Schema**

* **Manage Registration Tables:**
* tblStaff\_Registration (staff\_id[PK] , staff\_name ,staff\_ Gender ,staff\_ contact number ,staff\_ Email\_ID , password )
* FD: staff\_id[PK] {staff\_name,staff\_Gender, staff\_contact \_number, password}
* tblStudent (Student\_id[PK], S\_Nam,S\_Enrollment\_number,S\_Gender, S\_Semester, S\_division, S\_Course, S\_Contact Number , S\_Email\_Id , password )
* FD: Student\_id[PK] {S\_Enrollment\_number, S\_Name,S\_ Semester, S\_division, S\_course,S\_ Gender,S\_ Email\_Id , contact\_number, password}
* tblStudent\_Event\_Registration (S\_ID[PK] Erollment\_number,Occsion\_Name,Event\_name)
* FD: S\_ID[PK] {Enrollment\_number , Name, Gender,semester,division,course,Contact\_number,Email\_Id, Occasion\_Name,Event\_name}
* **Common Tables:**
* TblSub\_Event(subEvent\_Id[PK],Occasion\_Name, event\_name, event\_date, event\_time, event\_venue, Event\_Judge, Event\_Coordinator)
* FD:subevent\_id[PK]{occasion\_name,event\_name, event\_date, event\_time, event\_venue, Event\_Judge, Event\_Coordinator)
* **Result Tables:**
* TblEvent\_Result(r\_id[PK],Occasion\_Name,Event\_Name,Event\_Date,Student\_Enrollment\_Number,Rank)
* FD:r\_id[PK]{Occasion\_Name,Event\_Name,Event\_Date,Student\_Enrollment\_Number,Student\_Rank,Student\_Name,Student\_Gender,Student\_Course,Student\_Semester,Student\_Division,Student\_Contact,Student\_Email\_id}

**6.1.2 Data Dictionary**

tblStaff\_Registration:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Datatype** | **Size** | **Contain** | **Description** |
| Staff\_ID | Integer | 11 | Primary key, auto increment | Staff Id of the staff member |
| Staff\_Name | Varchar | 20 | Not null | Name of the staff member |
| Staff\_Gender | Char | 1 | Not null | gender of the staff member |
| Staff\_Contact\_Number | Bigint | 10 | Not null | Contact Number of the staff member |
| Staff\_Email\_ID | varchar | 20 | Not null | Email id of the staff member |
| Password | Varchar | 20 | Not null | Password of the staff member |

tblStudent\_Registration:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Datatype** | **Size** | **Contain** | **Description** |
| Student\_ID | Integer | 11 | Primary key, auto increment | Student Id of the student |
| S\_Name | Varchar | 20 | Not null | Name of the student |
| S\_Enrollment\_Number | Bigint | 15 | Not null | Enrollment Number of the student |
| S\_Gender | Char | 1 | Not null | Gender of the student |
| S\_Semester | Interger | 2 | Not null | Semester of the student |
| S\_Division | Char | 1 | Not null | Division of the student |
| S\_Course | varchar | 20 | Not null | Course of the student |
| S\_Contact\_Number | Bigint | 10 | Not null | Contact Number of the student |
| S\_Email\_Id | Varchar | 20 | Not null | Email id of the student |
| Password | varchar | 20 | Not null | Password of the student |

tblSub Event:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Datatype** | **Size** | **Contain** | **Description** |
| SubEvent\_Id | Integer | 11 | Primary key, Auto increment | Id of the sub event |
| Occasion\_Name | Varchar | 20 | Not null | Occasion Name of the event |
| Event\_Name | Varchar | 20 | Not null | Event to be held |
| Event\_Date | Date | - | Not null | Date of the event |
| Event\_Time | time | - | Not null | Time of the event |
| Event\_Venue | Varchar | 20 | Not null | Place of the event to be held |
| Event\_Cordinator | Varchar | 20 | Not null | Co-ordinator of the sub event |
| Event\_Judge | Varchar | 20 | Not null | Judge of the event |

tblStudent\_Event\_Registration:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Datatype** | **Size** | **Contain** | **Description** |
| S\_ID | Integer | 11 | Primary key, Auto increment | Id of the student event registration |
| Enrollment\_Number | Bigint | 15 | Not null | Enrollment Number of the student |
| Name | Varchar | 20 | Not null | Name of the student |
| Gender | Char | 1 | Not null | Gender of the student |
| Semester | Number | 1 | Not null | Semester of the student |
| Division | Char | 1 | Not null | Division of the student |
| Course | Varchar | 20 | Not null | Course of the student |
| Contact\_Number | Bigint | 10 | Not null | Contact Number of the student |
| Email\_Id | Varchar | 20 | Not null | Email id of the student |
| Occasion\_Name | Varchar | 20 | Not null | Occasion Name of the event |
| Event\_Name | varchar | 20 | Not null | Event to be held |

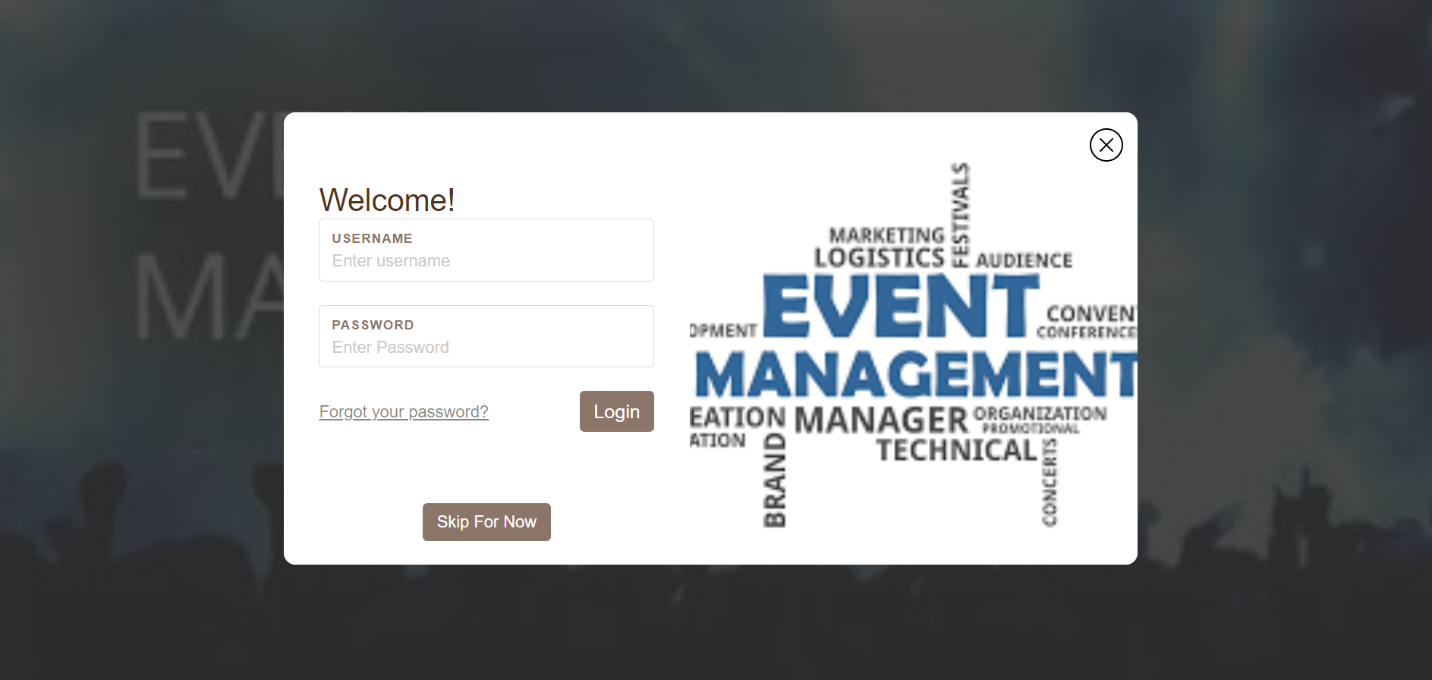
tblEvent Result:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Datatype** | **Size** | **Contain** | **Description** |
| R\_Id | Integer | 20 | Primary key, Auto increment | Id of the event result |
| Occasion\_Name | Varchar | 20 | Not null | Occasion Name of the event |
| Event\_Name | varchar | 20 | Not null | Event to be held |
| Event\_Date | Date | - | Not null | Date of the event |
| Student\_Enrollment | Bigint | 15 | Not null | Enrollment Number of the student |
| Student\_Rank | Integer | 1 | Not null | rank of the student in the event |
| Student\_Name | Varchar | 20 | Not null | Name of the student |
| Student\_Gender | Char | 1 | Not null | Gender of the student |
| Student\_Course | varchar | 20 | Not null | Course of the student |
| Student\_Semester | Interger | 2 | Not null | Semester of the student |
| Student\_Division | Char | 1 | Not null | Division of the student |
| Student\_Contact | Bigint | 10 | Not null | Contact Number of the student |
| Student\_Email\_Id | Varchar | 20 | Not null | Email id of the student |

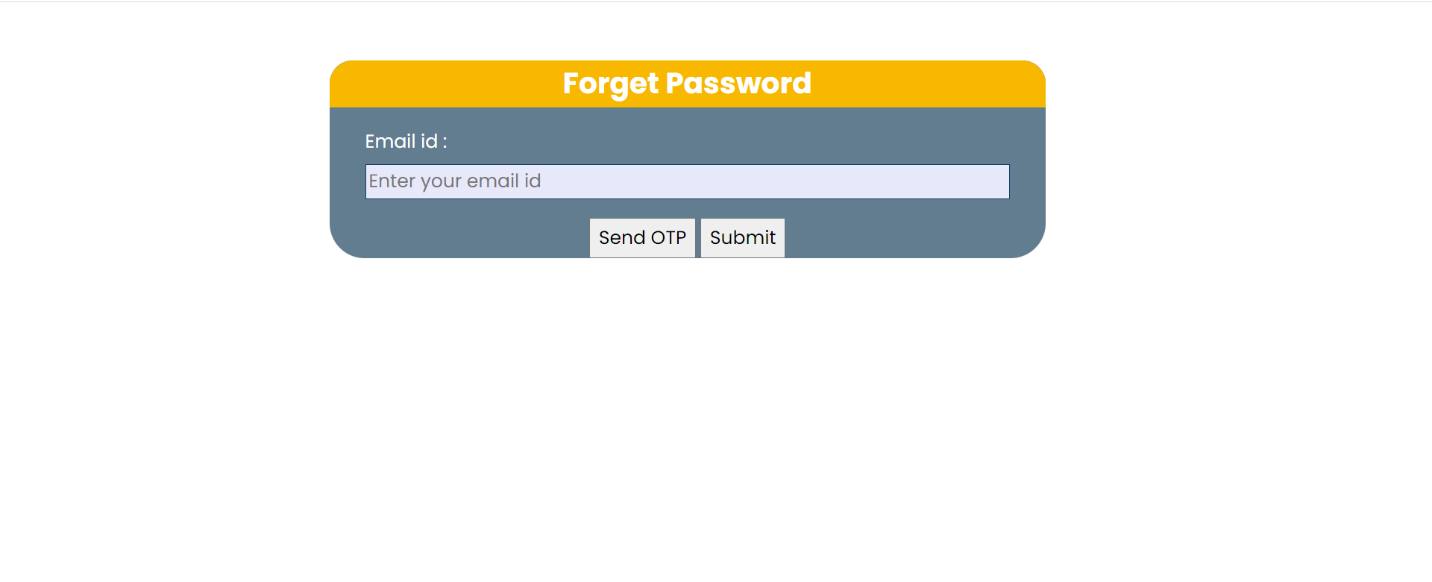
**7.System Implementation**

**7.1 Screenshots**

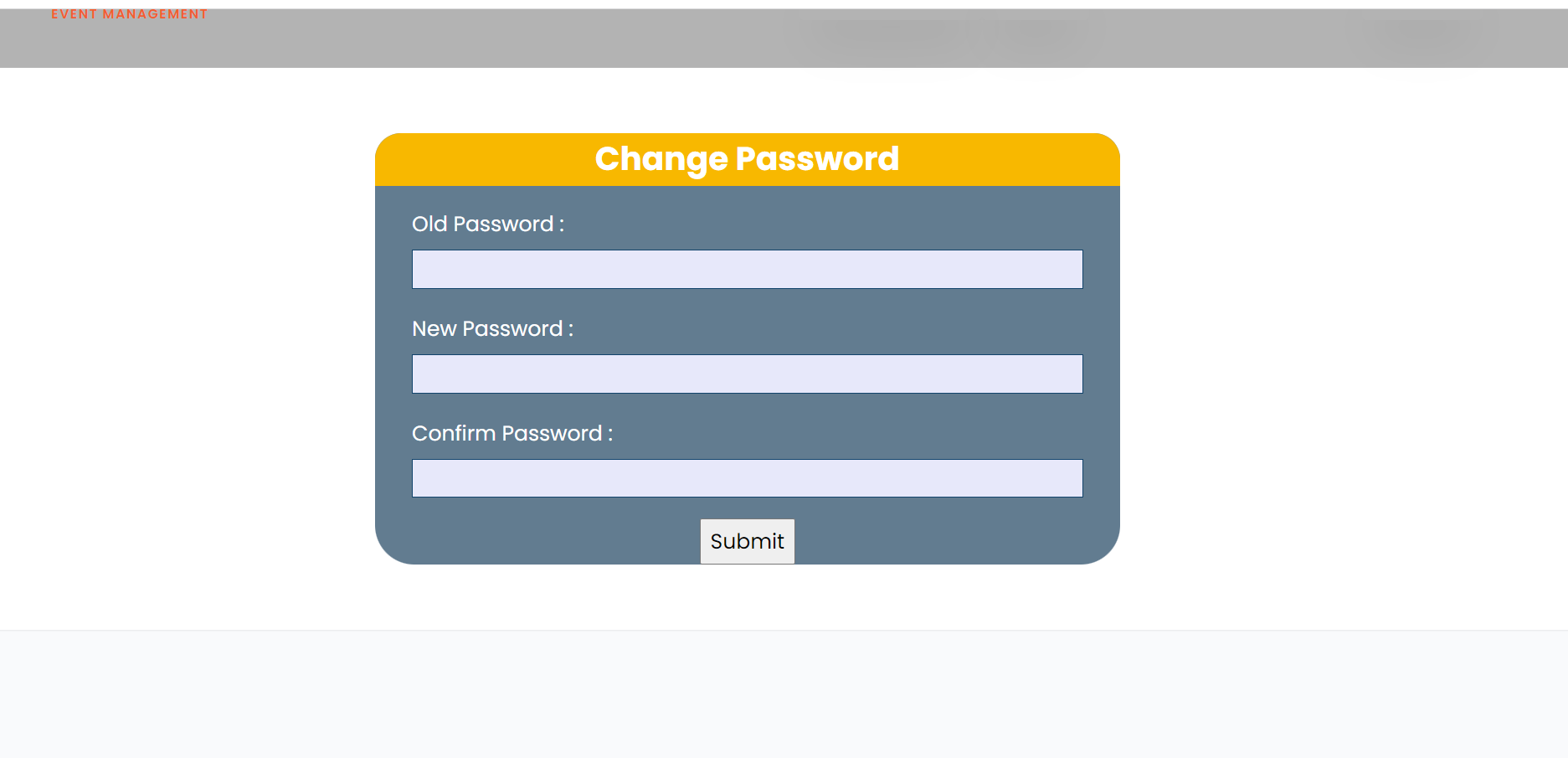
Login Page



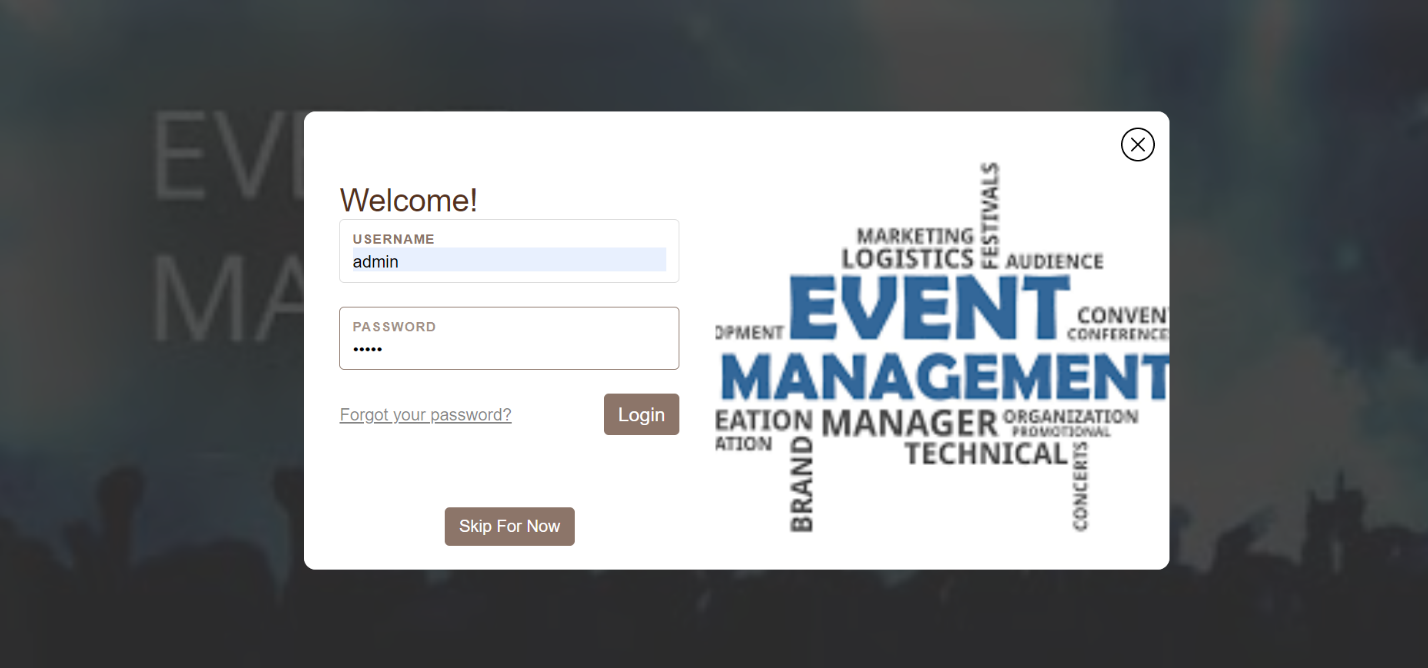
Forget Password



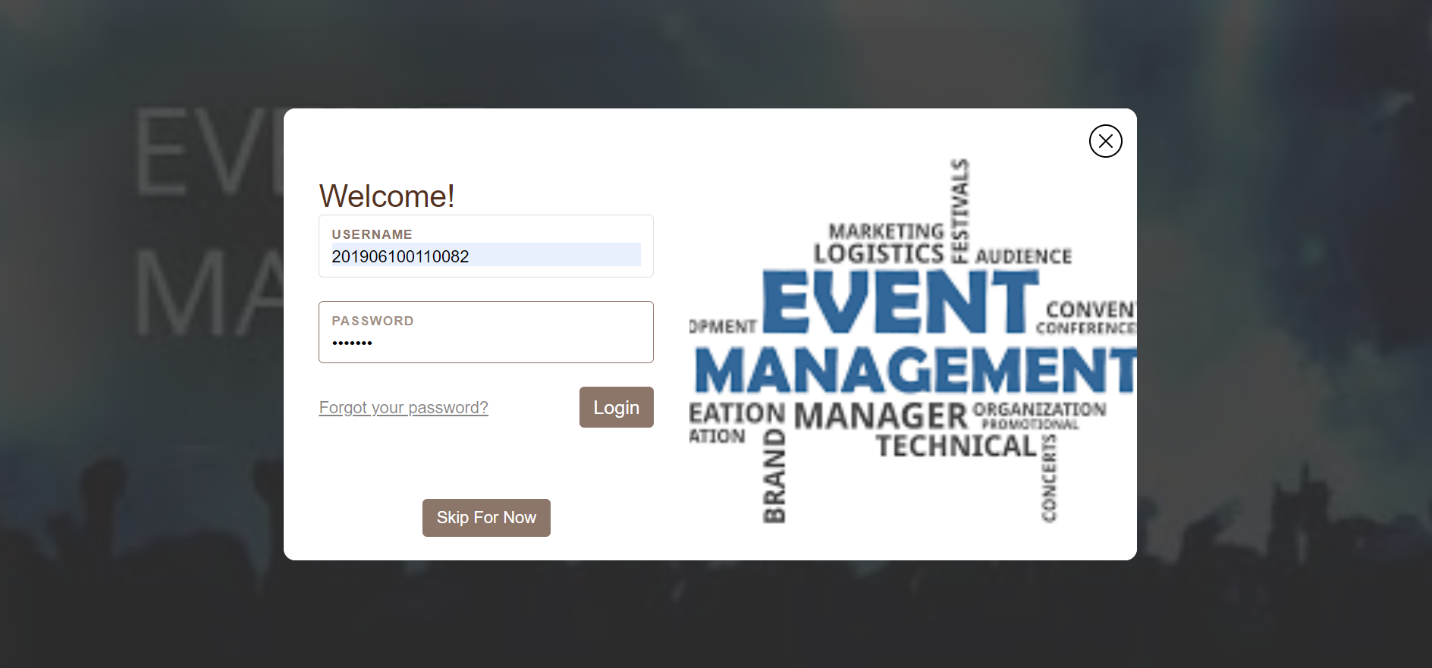
Change Password



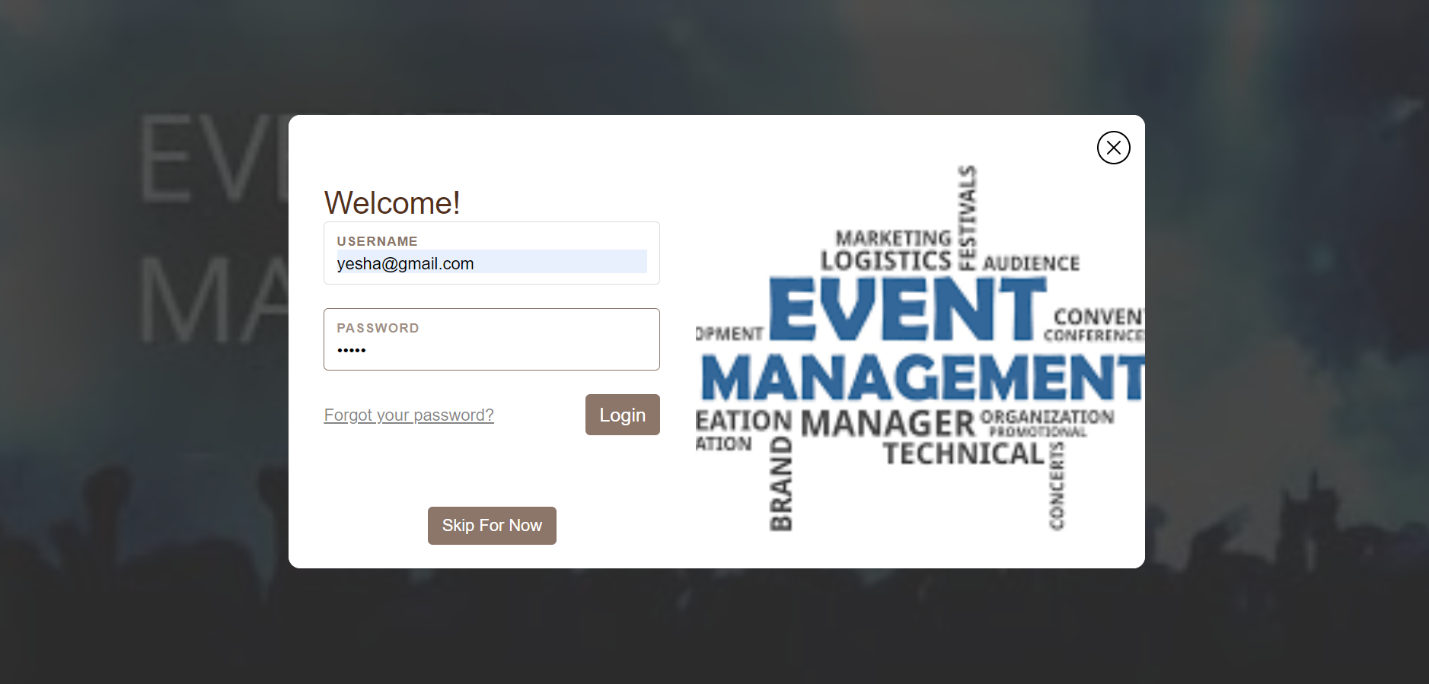
Admin Login



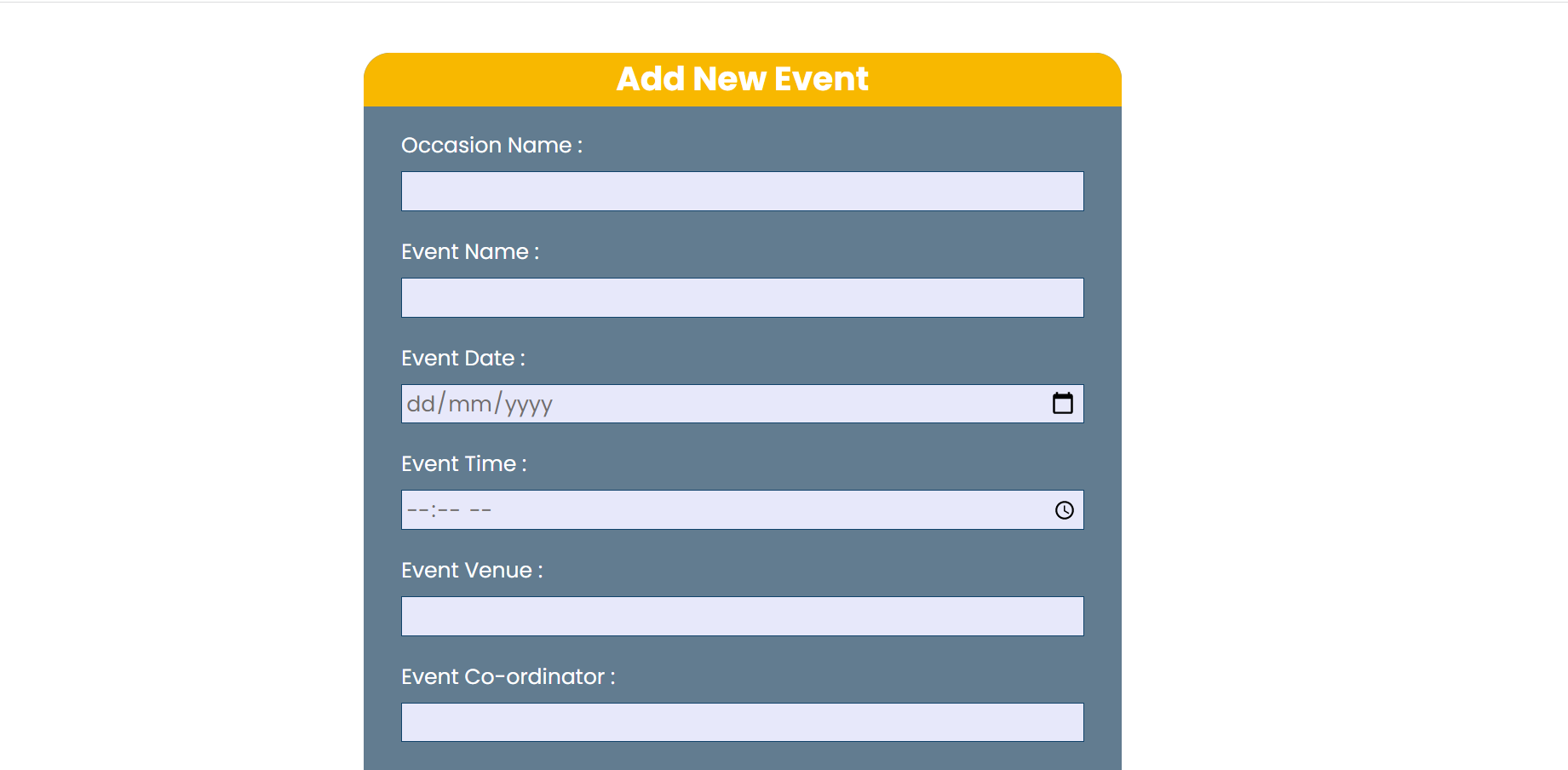
Student Login



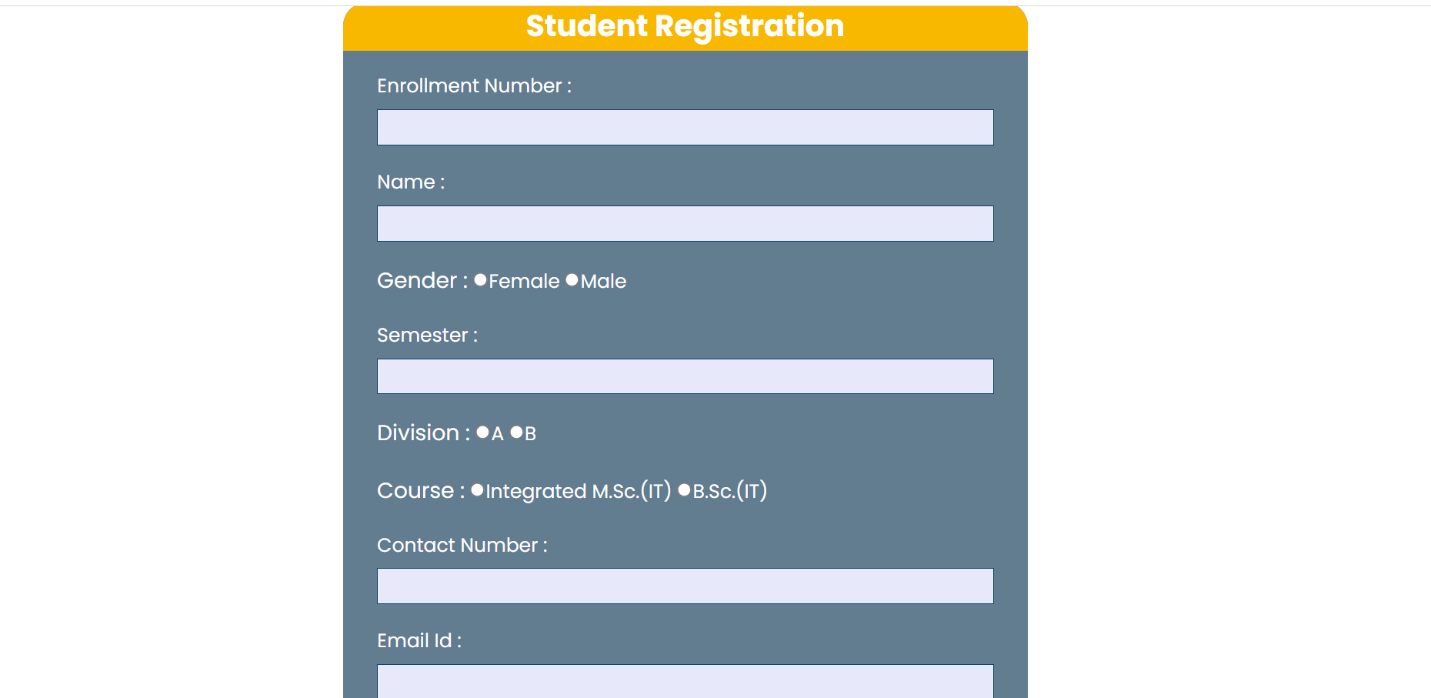
Staff Login



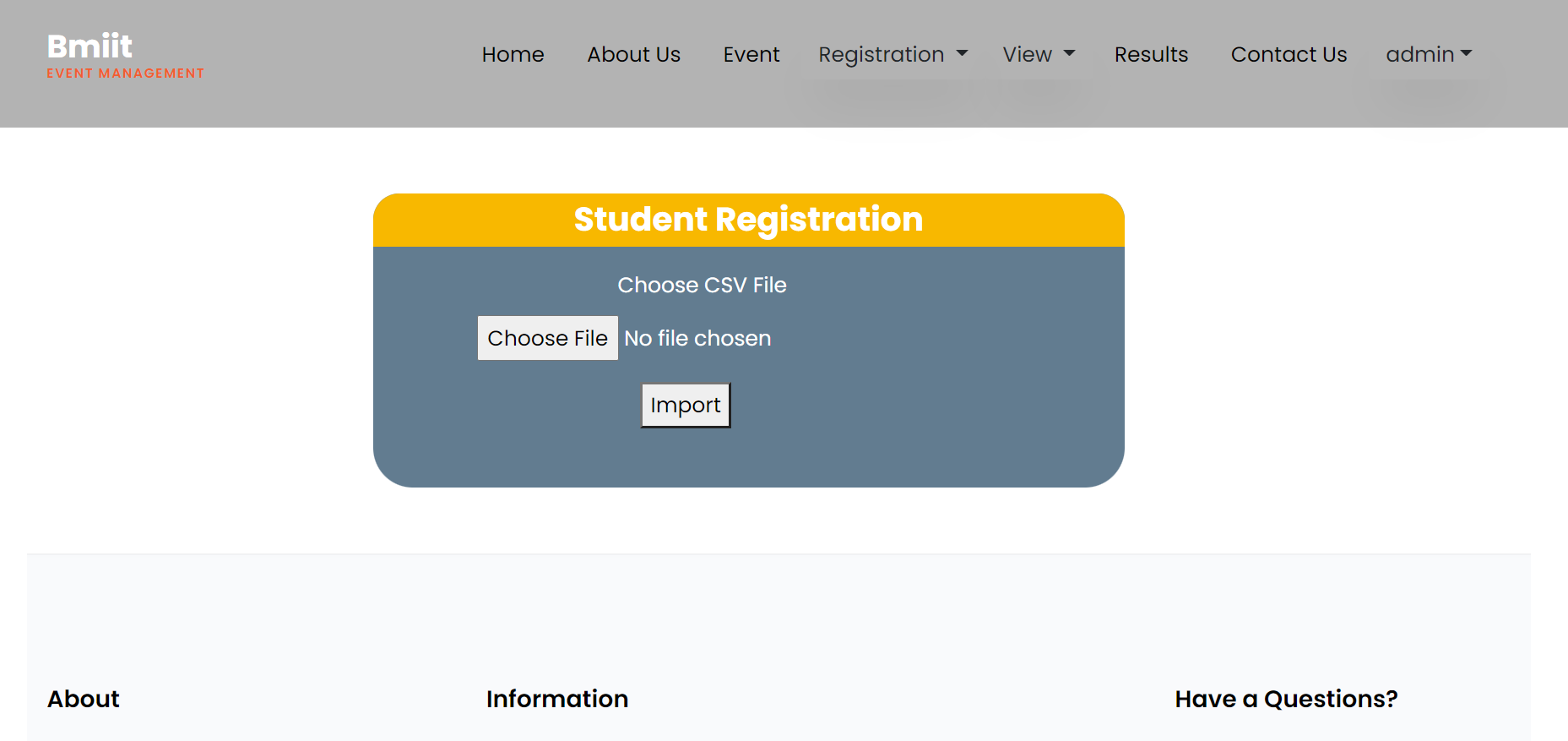
Add New Event



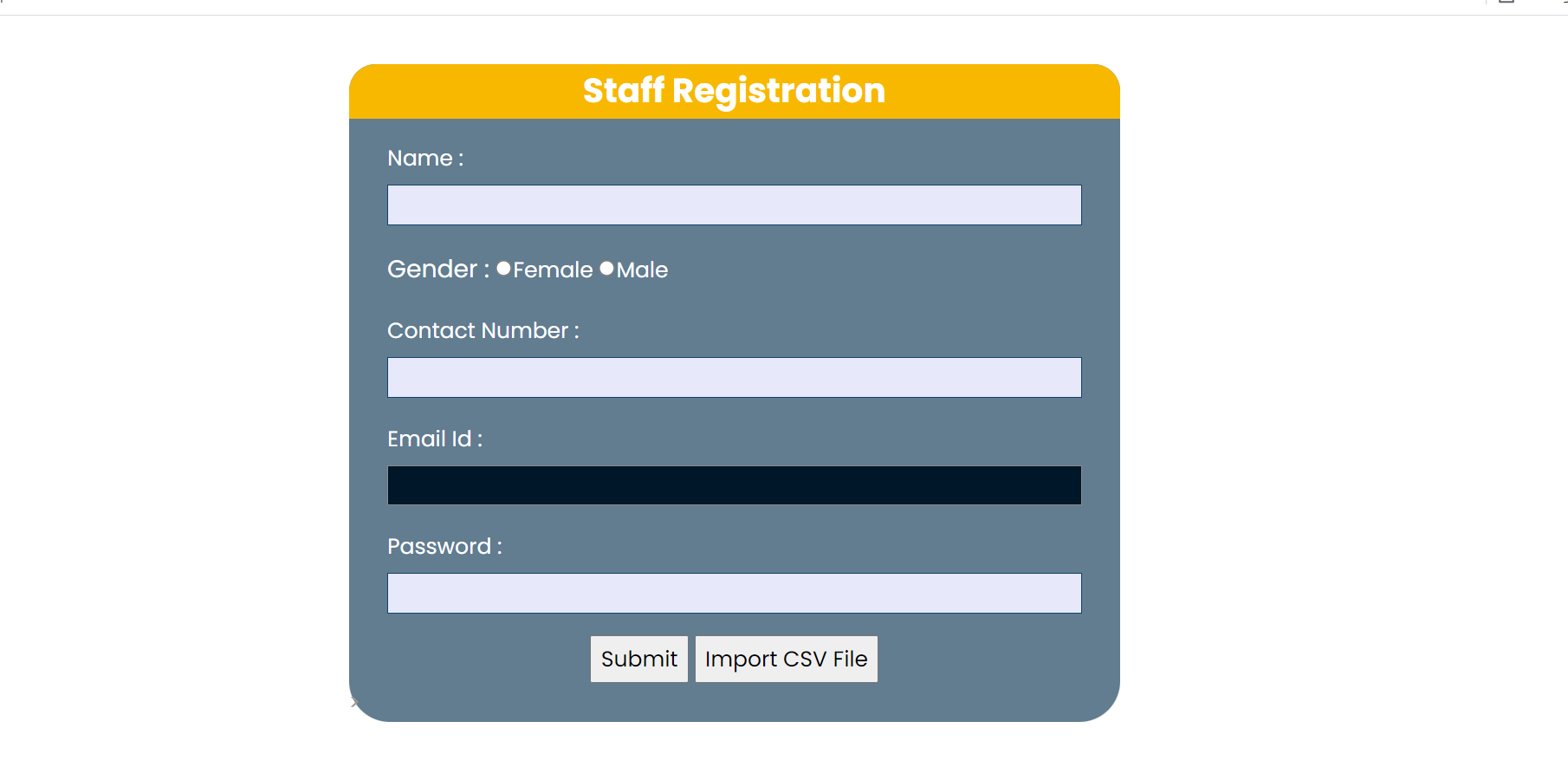
Student Registration



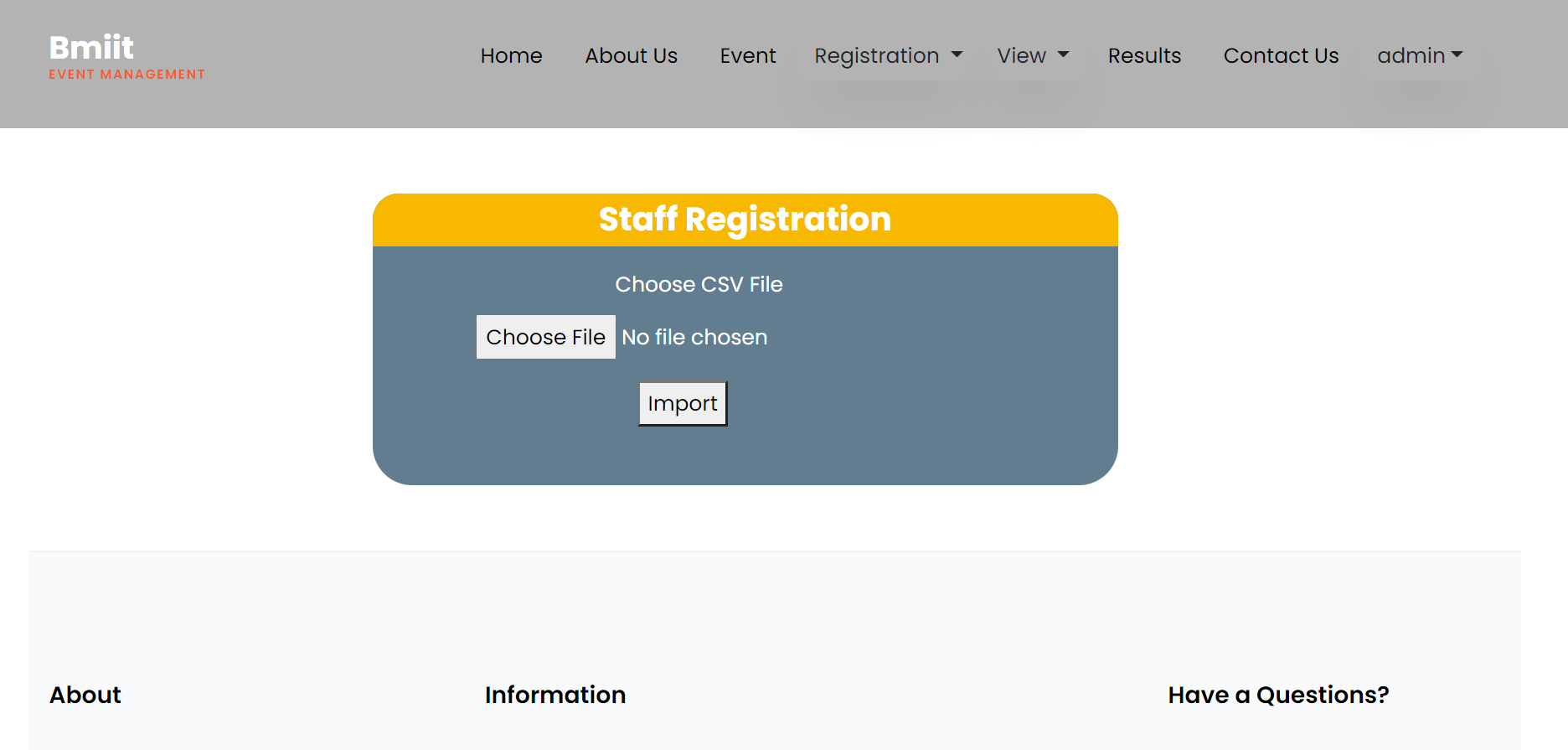
CSV file upload of student



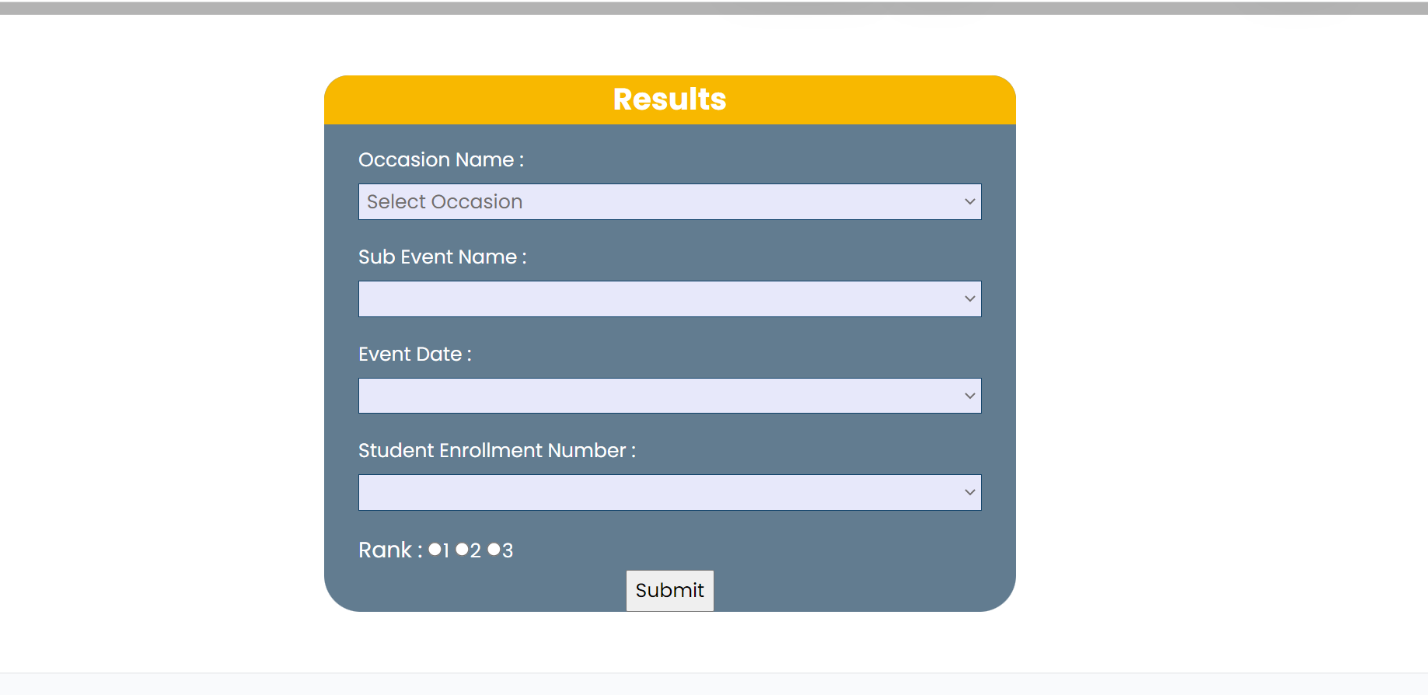
Staff Registration



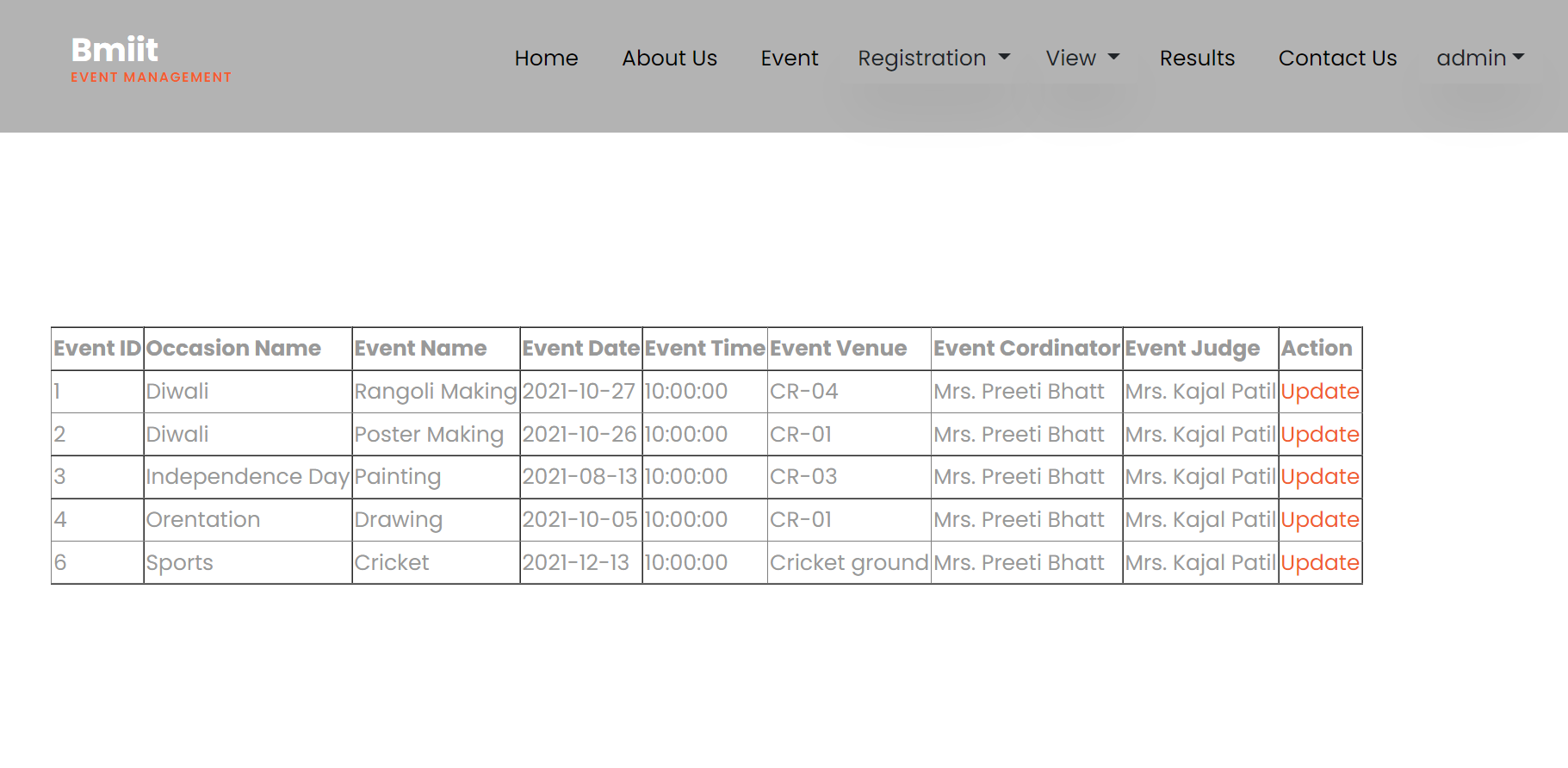
CSV file upload for staff



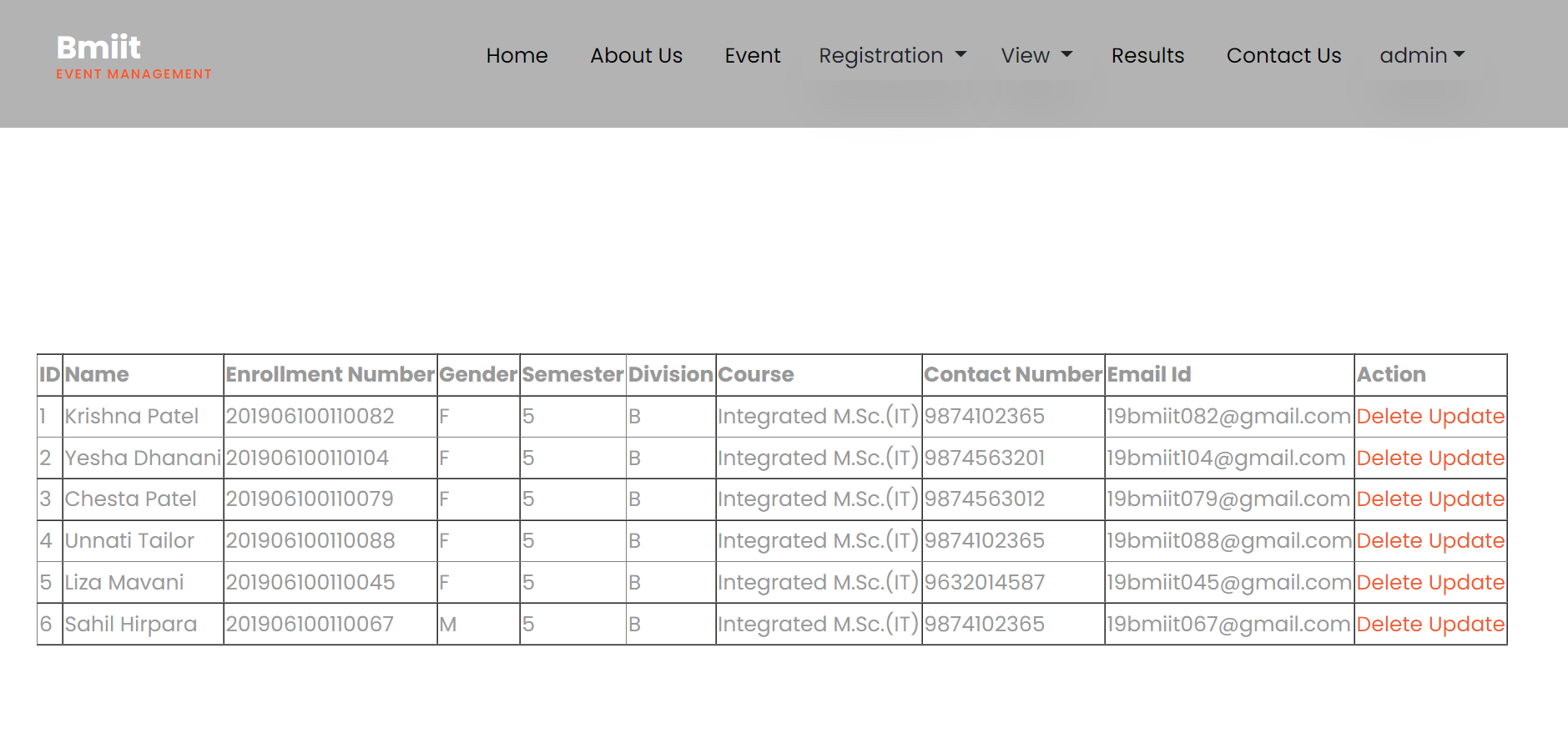
Upload Result



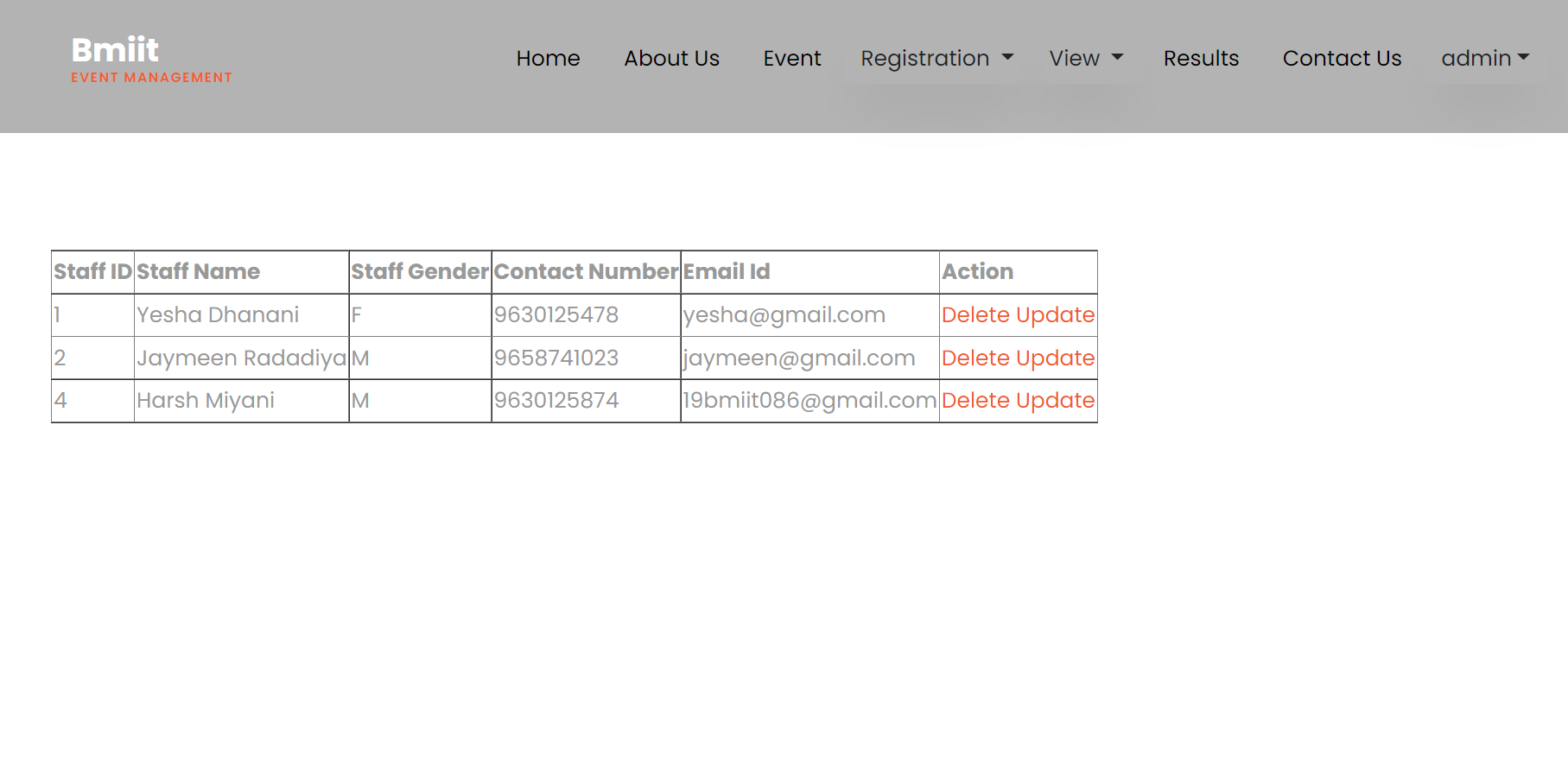
View /update Event



View, update/delete Student



View, update/delete Staff



**8 Testing**

8.1 Test Cases

Test Case for Event

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input1**  **Occasion\_Name** | **Input2**  **Event\_Name** | **Input3**  **Event\_Date** | **Input4**  **Event\_Time** | **Input5**  **Event\_Venue** | **Input6**  **Event\_Cordinator** | **Input7**  **Event\_Judge** | **Expected Result** | **Remark(if any)** |
| TC1 | Add Event | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Event added successfully | - |
| TC2 | Occasion name is invalid | InValid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Event not added | Occasion name not in proper format |
| TC3 | Event Name is invalid | Valid Input | InValid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Event not added | Event name is not in proper fromat |

Test Case for Student

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input1**  **Enrollment\_Number** | **Input2**  **Student\_Name** | **Input3**  **Student\_Gender** | **Input4**  **Semester** | **Input5**  **Division** | **Input6**  **Course** | **Input7**  **Contact\_number** | **Input8**  **Email\_id** | **Input9 Password** | **Expected Result** | **Remark(if any)** |
| TC1 | Add Student | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Student added successfully | - |
| TC2 | Enrollment is invalid | InValid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Student not added | Enrollment not in proper format |
| TC3 | Student Name is invalid | Valid Input | InValid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Student not added | Student name is not in proper fromat |
| TC4 | Contact Number is invalid | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | InValid Input | Valid Input | Valid Input | Student not added | Contact number is not in proper format |
| TC5 | Email id is invalid | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | InValid Input | Valid Input | Student not added | Email id is not in proper format |
| TC6 | Password is invalid | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | InValid Input | Sudent not added | Password is not in proper format |

Test Case for Staff

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input1**  **Name** | **Input2**  **Gender** | **Input3**  **Contact\_Number** | **Input4**  **Email\_id** | **Input5 Password** | **Expected Result** | **Remark(if any)** |
| TC1 | Add Staff member | Valid Input | Valid Input | Valid Input | Valid Input | Valid Input | Staff member added successfully | - |
| TC2 | Name is invalid | InValid Input | Valid Input | Valid Input | Valid Input | Valid Input | Staff member not added | Name is not in proper fromat |
| TC3 | Contact Number is invalid | Valid Input | Valid Input | InValid Input | Valid Input | Valid Input | Staff member not added | Contact number is not in proper format |
| TC4 | Email id is invalid | Valid Input | Valid Input | Valid Input | InValid Input | Valid Input | Staff Member not added | Email id is not in proper format |
| TC5 | Password is invalid | Valid Input | Valid Input | Valid Input | Valid Input | InValid Input | Staff member not added | Password is not in proper format |

Conclusion

Institute Event Management System is friendly and effective system. The system is customized with the management of the event, student and student participation. The main purpose of the Institute Event Management System is to reduce the paper work and make the event information available easily to the student and members of the institute. Through the system maintain the event participation becomes easy.

**Bibliography**

* Steven Holzner, PHP : THE COMPLETE REFERENCE, Indian Edition
* [www.phppot.com](http://www.phppot.com)
* [www.greeksforgreeks.com](http://www.greeksforgreeks.com)
* <https://getbootstrap.com/>