**BMS COLLEGE OF ENGINEERING**

**(Autonomous College under VTU)**

**Bull Temple Road, Basavanagudi, Bangalore – 560019**



A project report on

***“BOOK MY EVENT @BMSCE”***

Submitted in partial fulfillment of the requirements for the award of degree

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION SCIENCE AND ENGINEERING**

By

Ankith B Rao, 1BM17IS011

H.Anarghya, 1BM17IS028

Lalitashree R.H, 1BM17IS039

Pannaga Sharma M.L, 1BM17IS053

Under the guidance of

Dr.Shambhavi B.R, Associate Professor

**Department of Information Science and Engineering**

**Aug-Dec 2019**

**BMS COLLEGE OF ENGINEERING**

**(Autonomous College under VTU)**

**Bull Temple Road, Basavanagudi, Bangalore – 560019**



**Department of Information Science and Engineering**

CERTIFICATE

This is to certify that the project entitled “TITLE OF THE PROJECT” is a bona-fide work carried out by **Ankith B Rao(1BM17IS011), H.Anarghya(1BM17IS028), Lalitashree R.H(1BM17IS039), Pannaga Sharma M.L(1BM17IS053)** in partial fulfillment for the award of degree of Bachelor of Engineering in **Information Science and Engineering**  from **Visvesvaraya Technological University, Belgaum** during the year **2019-2020**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

**Signature of the Faculty Signature of the HOD**

**Name and Designation Name and Designation**

Table of Contents

Page No.

Acknowledgements

Abstract

1. **Introduction ……………………………………………………………………………….**
   1. Purpose ………………………………………………………………………………
   2. Scope …………………………………………………………………………………
2. **Software Requirement Specification …………………………………………………….**
   1. Software Requirements ……………………………………………………………….
   2. Hardware Requirements ………………………………………………………………
   3. Functionality……………………………………………………………………………
3. **ER Diagram ……………………..………………………………………………………..**
4. **Relational Schema & Normalization ……………………………………………….……**
5. **Implementation……………………………………………………………………………**
   1. SQL queries ………………………………………………………………………….
   2. Connecting backend to frontend ……………………………………………………….
6. **Conclusion**………………………………………………………………………………..
   1. Summary………………………………………………………………………………..
   2. Limitations……………………………………………………………………………..
   3. Further enhancements…………………………………………………………………..
7. **References**……………………………………………………………………………..
8. **Appendix: Snapshots** …………………………………………………………………

**INSTRUCTIONS:**

1. Use times new roman text – 12 size
2. Title headings bold with -16 size
3. Sub title bold with -14 size
4. Report should be minimum of 25 pages
5. Report to be spiral bound with blue transparency

**ABSTRACT**

In this modern era where everything is going online, why not make the tideous process of registering for any event happening in college also go online. The application developed allows the students to find their events of interest and register for the same using the application. In this college where there are so many events happening it is difficult to keep track of all of them on paper. This application provides a user friendly user interface for everyone to use and also handles the database. The application also lets the user know the different student clubs and their events present in the college. This online registration saves a lot of time and work since there is not much involvement of people in this. The ultimate goal of the application is to reduce the work and time of people involved.

**ACKNOWLEDGEMENTS**

We as a team, would like to express our gratitude to all those people who has helped us throughout the project for the completion of the project. The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Dr.Shambhavi, for providing us an opportunity to do the project work in the college and giving us all support and guidance which made us complete the project duly. We are extremely thankful to her for providing such a nice support and guidance.

We would also like to express gratitude to each other, because as a team, we were able to divide the work among ourselves so that it is not too much workload for anyone and everyone in the team has co-operated well for the completion of this project. Working on this has made us learn all the new things and do a lot of research in order to make this project work. Hence we would like to express our thanks of gratitude to everyone who has helped us in this.

**INTRODUCTION**

**Purpose:**

To design and develop an application for online booking and registration of various events held at BMSCE during the fests, by the numerous student clubs and departments present in BMSCE. The application includes an admin login in which he/she can update, add or remove the events as the organizers of the events wish. It also includes student user interface where students or anyone who wish to participate can look for the events of their interest and register for the same.

The application is built considering the class of people who register for events in college, i.e., students. It is made sure that they do not face any difficulty while using the application. The application consists of two parts, the user interface for students and the user interface for the admins. The admins of each club are allowed to login with their respective credentials and have access to add, update or delete any event related to their club. They also have the access to view the registrations. There is minimum of one coordinator for each event who is also handled by the admin. In the student user interface, the students can register for any of the events and an email notification of the same is sent to the student. The student can view the various events being held and then register. They have the option of either paying through cash or card. The user interface is designed such that it is easy for anyone who wish to register to make registrations for the event they desire to participate.

**Scope:**

Since everything these days is done over the internet, there are less chances that for works like these will be done on paper in the near future. Everything is dependent on technology nowadays and it would be advantageous if registrations are also made to go online which will save a lot of work.

**REQUIREMENT SPECIFICATION**

**Software requirements:**

1. MYSQL Server
2. JDBC driver

**Hardware requirements:**

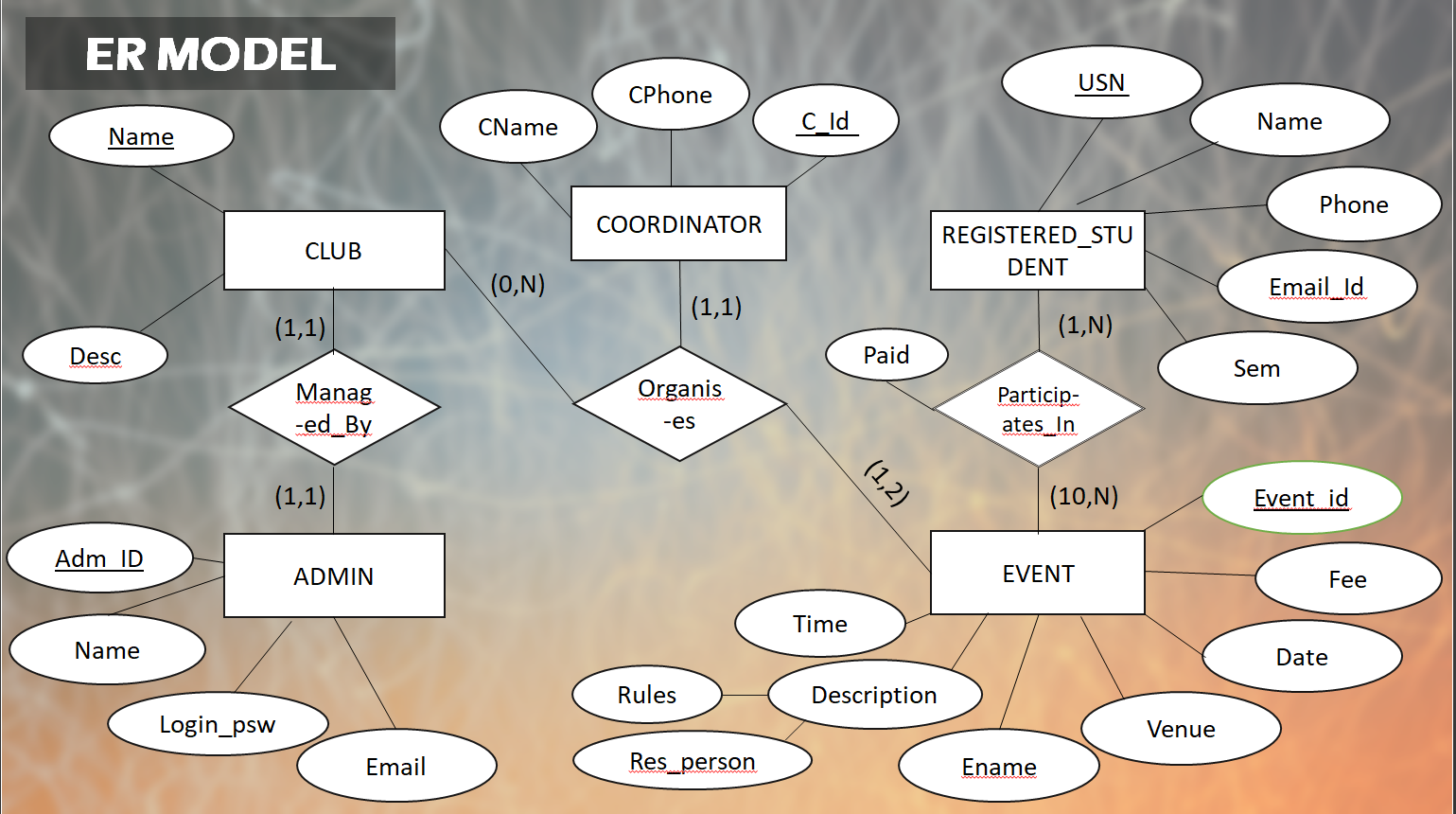
1. Functional Operating System
2. Processor – core i5/i7

**Functionality:**

1. The various events are organized and handled by different clubs. Each club has a unique name. Every club is managed by an admin who can login and edit any details regarding their club.
2. Each club handles one or more events and for each event, we store the event number, event name, the date, the venue at which it will be held.
3. We store the details of each student who have registered for a particular event. We store their USN, name, semester to which they belong and their contact details.
4. The admin will have the access to update, modify, add or delete events. He/she will be responsible for the handling of the database.
5. The user/student will be able to register for the events. He/she will be able to enter their complete details and details of the event they want to participate.
6. After each registration by the students , a confirmation message and the details will be sent to the mail of the student.

**ER DIAGRAM**

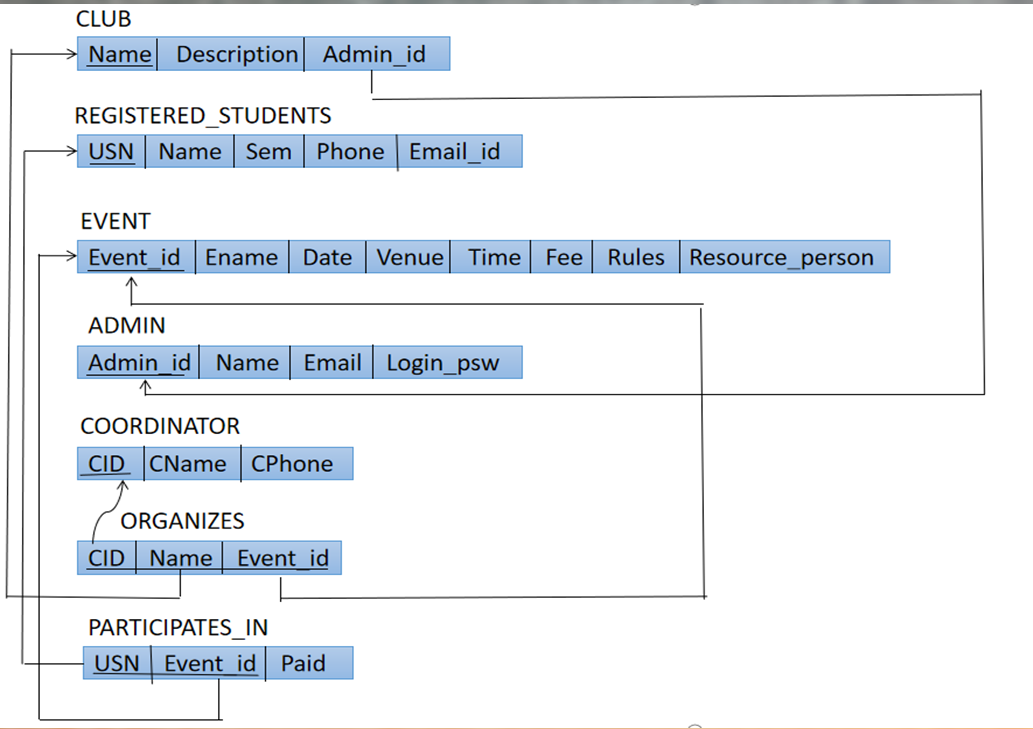
The Entity-Relationship diagram drawn as per the requirements of the database is as follows:



**RELATIONAL SCHEMA AND NORMALIZATION**

The relational schema is obtained by mapping the ER diagram into corresponding tables(relations), it is a graphical representation of how the database is organized into tables. A relation schema R of degree N is denoted by R(A1,A2,A3….An), where A1, A2, A3 …An are attributes.

The relational schema is as follows:



The underlined attributes denote the primary key and the foreign keys are denoted by arrows. All schema based constraints like not null, domain constraint, entity and referential integrity constraints are added.

**Normalization:**

Normal form of a relation refers to the highest normal form condition that it meets and hence indicates the degree to which it has been normalized. The schema is decomposed into smaller relation schemas that contain the subset of attributes of the main schema and still hold the functional dependencies.

There are totally five normal forms , but for our project we have considered up to 3NF.

1NF : It states that the domain of an attribute must include only atomic values and the value of any attribute in a tuple must be single value, i.e., there should be no multivalued attributes.

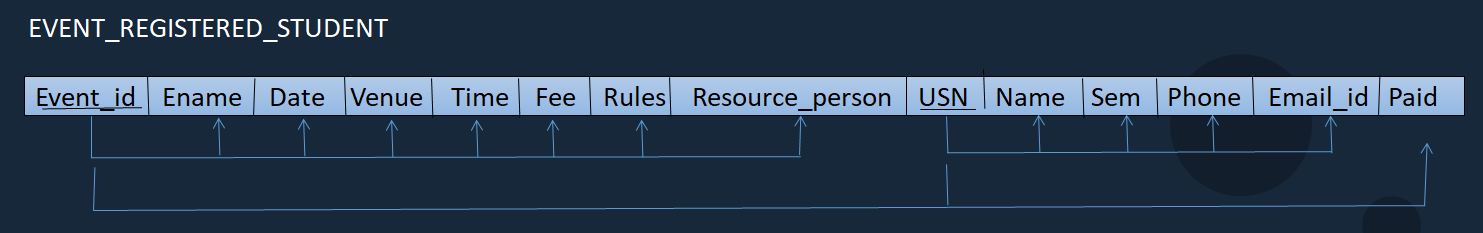
2NF : It states that every non-prime attribute in the schema should be fully functionally dependent on the primary key.

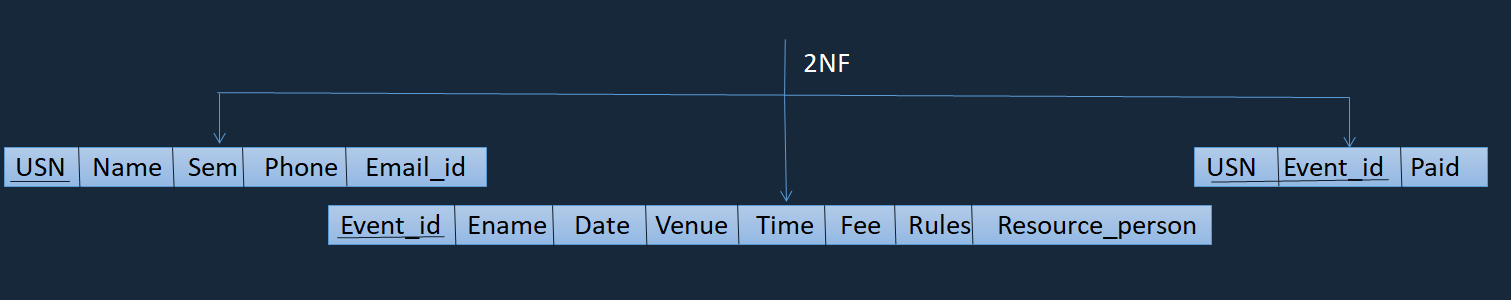
3NF : It states that the schema should be in 2NF and no non prime attribute should be transitively dependent on the primary key.

There are other higher normal forms too like, Boyce-Codd normal form, fourth and fifth normal forms which are not considered for our project.

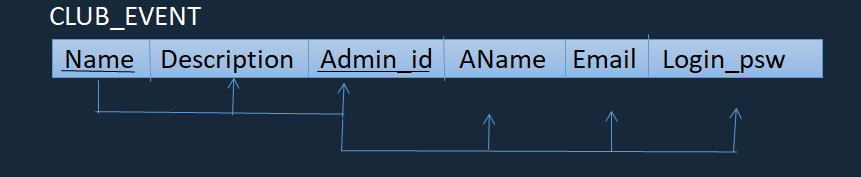
The schema is by default in normal form when it is mapped from the ER diagram. To demonstrate normalization, relations have been merged and then decomposed. The normalization of the schema is as shown:

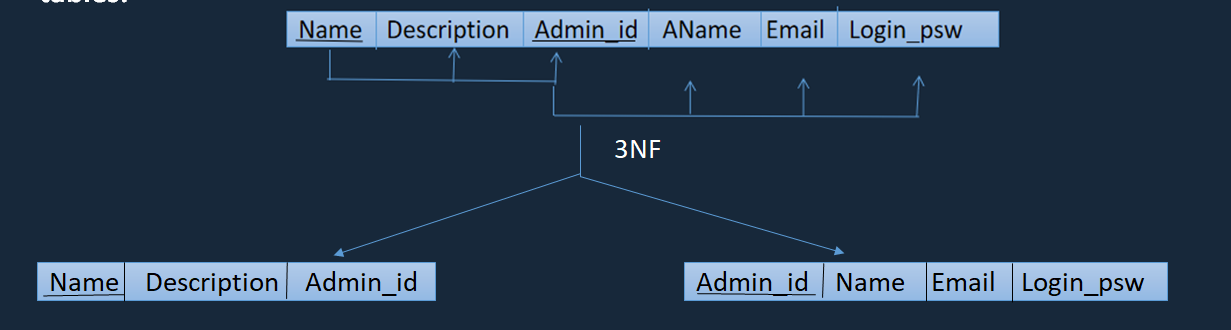
The table considered below is the event and registered student table combined. Since there are no multivalued attributes involved, it is already in 1NF. But all the nonprime attributes are not fully functionally dependent on the primary key of the table, hence it is not in 2NF. To bring it to 2NF ,it is decomposed into three different tables, where the functional dependency is not lost and there , it is made sure that all the nonprime attributes are fully functionally dependent on the primary key only. The tables are in 3NF since there are no transitive dependencies.





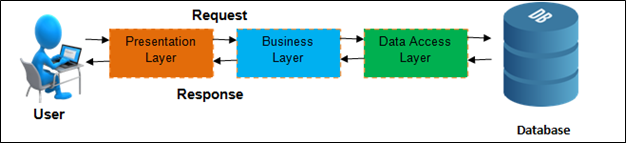
Now, consider the club and event table combined, it is already in 1NF as there are no multivalued attributes. It is also in 2NF as every nonprime attribute is fully functionally dependent on the primary key. But there are transitive dependencies. To remove them, the table is decomposed into two new tables, preserving the functional dependencies. After the decomposition, it is in 3NF.





**IMPLEMENTATION**

**Architecture : Three tier architecture**



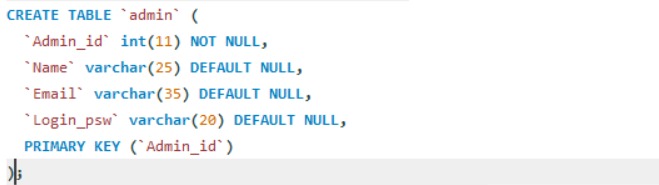
The three tier architecture shows a presentation layer which is represented by the user interface done with the help of Java. It is the part that interacts with the user.

It contains a Business logic layer between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user. The application layer(business logic layer) also processes functional logic, constraint, and rules before passing data to the user or down to the DBMS.

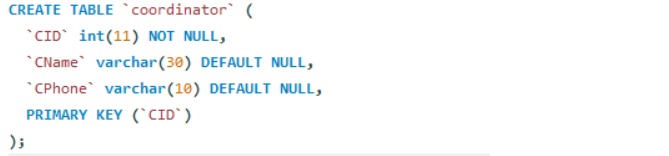
The layer that is in contact with the database is the data access layer. The connection between the database and the user interface is obtained using the JDBC API. The database stores the data and the metadata.

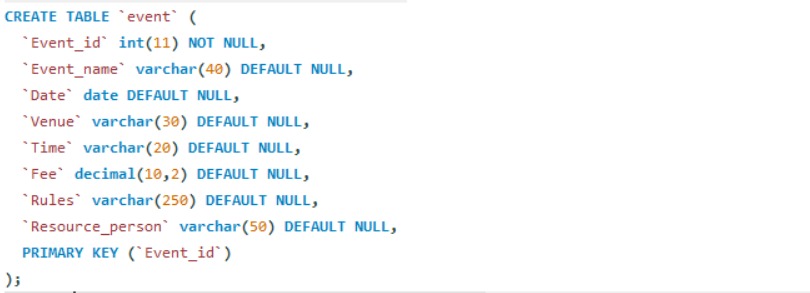
**SQL Queries:**

The main SQL command for data definition is the create statement which is used to create schemas , tables , types and domains , as well as other constructs such as views , assertions, triggers. The create commands for our project is as follows:

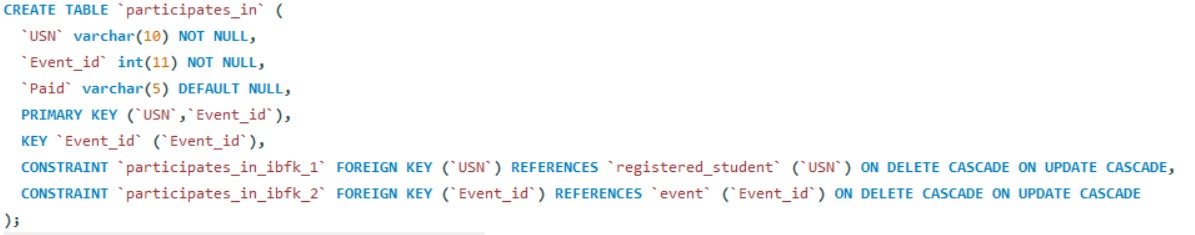


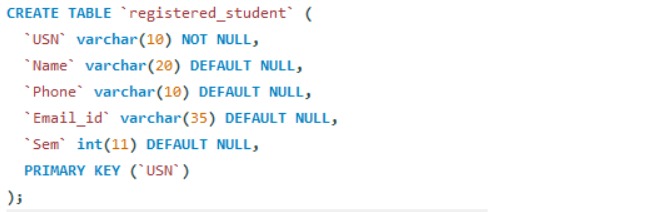












After the tables are created, they are to be populated. Insert statements insert values into the tables according to the order specified and it fails if the entered values do not match the domain of the attributes or if they are not in order. Then to fetch the data, Select statements are used which return the values which satisfy the condition given in the select statement.

Examples of insert and select statements are given below:

INSERT INTO `admin` VALUES (1,'Lalita','lalitashree.is17@bmsce.ac.in');

INSERT INTO `club` VALUES ('Chirantana','BMSCE Kannada Sangha ',4);

INSERT INTO `coordinator` VALUES (1,'Ravi','9945837567');

INSERT INTO `event` VALUES (1,'CODIGO','2020-04-14','ISE lab,New Building','10:00:00',100.00,'1.Individual Event 2.Consists of two rounds 3.The Judges decision is final',NULL);

INSERT INTO `organizes` VALUES (3,'ISE Student Club',1),(4,'ISE Student Club',2);

INSERT INTO ‘registered\_student’ VALUES ('1BM17IS023 ' , 'Gagana' , '8908907887' , 'gagana@gmail.com',5);

SELECT \* FROM event;

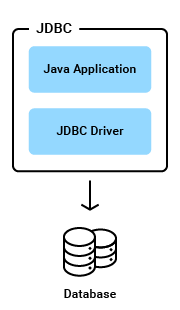
SELECT event\_id from event where event\_name=’codigo’;

UPDATE participates\_in set paid=’yes’ where name=’suresh’;

All these queries are executed in the MySql workbench where our schema and the tables are created. They are also written after connecting frontend to backend and when some value is to be fetched from the database to the user interface.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Connecting backend to frontend:**



Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity.It provides methods to query and update data in the database. And it is oriented towards relational databases.

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT.

There are 5 steps to connect any java application with the database using JDBC. These steps are as follows:

* Register the Driver class
* Create connection
* Create statement
* Execute queries
* Close connection

The sample code for connectivity is as shown below:



**CONCLUSION**

**Summary:**

The application is built using Swing toolkit in Java, which is the user interface of the application and the database is in the Mysql server and written in SQL. JDBC is used for the connectivity between the user interface and the database. The application is built considering the class of people who register for events in college, i.e., students. It is made sure that they do not face any difficulty while using the application. The application consists of two parts, the user interface for students and the user interface for the admins. The admins of each club are allowed to login with their respective credentials and have access to add, update or delete any event related to their club. They also have the access to view the registrations. There is minimum of one coordinator for each event who is also handled by the admin. In the student user interface, the students can register for any of the events and an email notification of the same is sent to the student. The student can view the various events being held and then register. They have the option of either paying through cash or card. The user interface is designed such that it is easy for anyone who wish to register to make registrations for the event they desire to participate.

Three tier architecture is being followed for the design of the application and proper constraints are laid on the data being entered to make sure that proper data is being stored in the database. Overall the application is for the students , to make them register for the events happening in college , easily and without making them search for the right place and people to register, this online platform makes it easier and saves time.

**Limitations:**

1. The application is designed keeping in mind only one semester. It could be extended or made to apply for all semesters.
2. The application focuses majorly on students and others who register, (not students) are not taken into consideration.
3. The dates of the events (past / present) are not clearly mentioned while the user registers, it is something that the user has to make sure that he/she does not register for an event which has been completed.

All these are limitations which we believe can be solved with more time on the project. Given a chance in the future, more efforts will be put to make this application a better one and made sure that every small issue in the application is taken care of.

**Future Enhancements:**

If given a chance to enhance this project, there are still validations to be added. All the limitations that are listed above will be solved and will make sure that the major validation with dates of the events will be added and the users will not have to face any problems or confusions while registering for any of the events. The providing of certificates to the participants can be added as a part of the project where in the registered student is confirmed to have attended the event and after the event finishes the certificates can be mailed to them. The user interface can be made more user friendly and understandable.

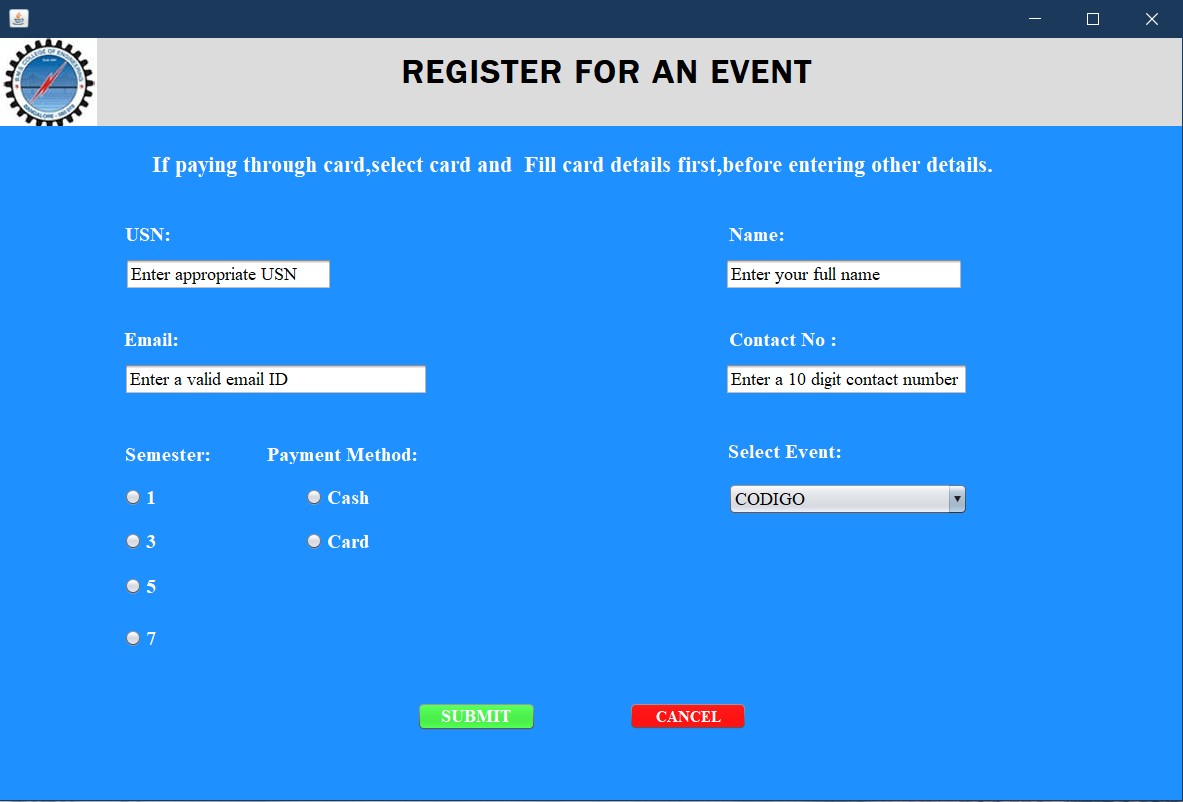
**References**

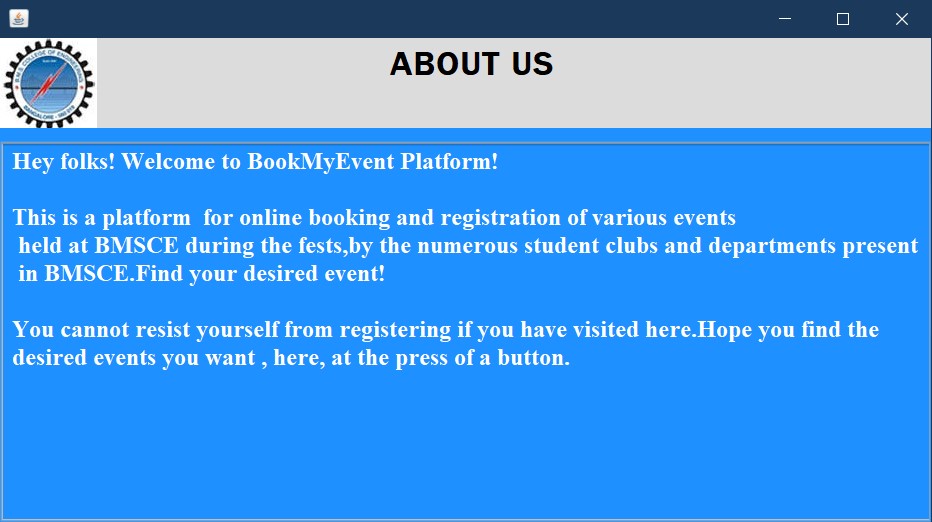
1. Java - The Complete Reference (JDK 8) by Herbert Schildt
2. Fundamental of Database Systems by Elmasri and Navathe
3. [JDBC Tutorial - Tutorialspoint](https://www.tutorialspoint.com/jdbc/index.htm)
4. [SWING - Overview - Tutorialspoint](https://www.tutorialspoint.com/swing/swing_overview.htm)
5. [Example of sending email in Java - javatpoint](https://www.javatpoint.com/example-of-sending-email-using-java-mail-api)

**Appendix : Snapshots**

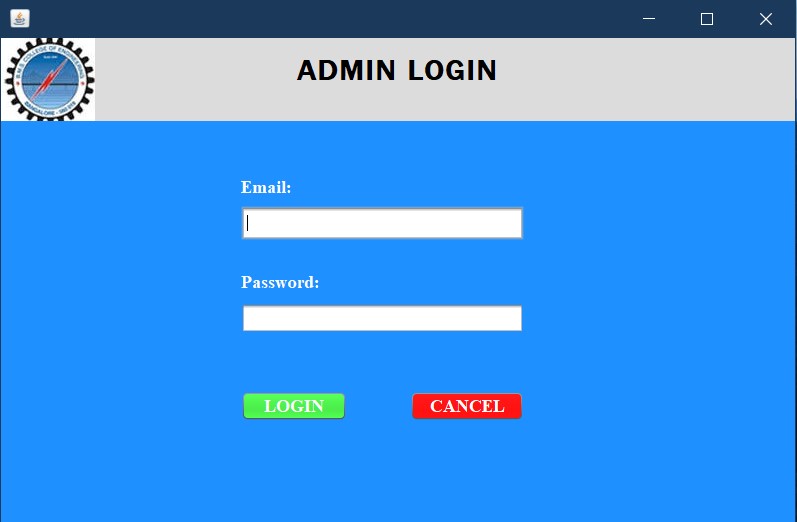
**1. Home page:**



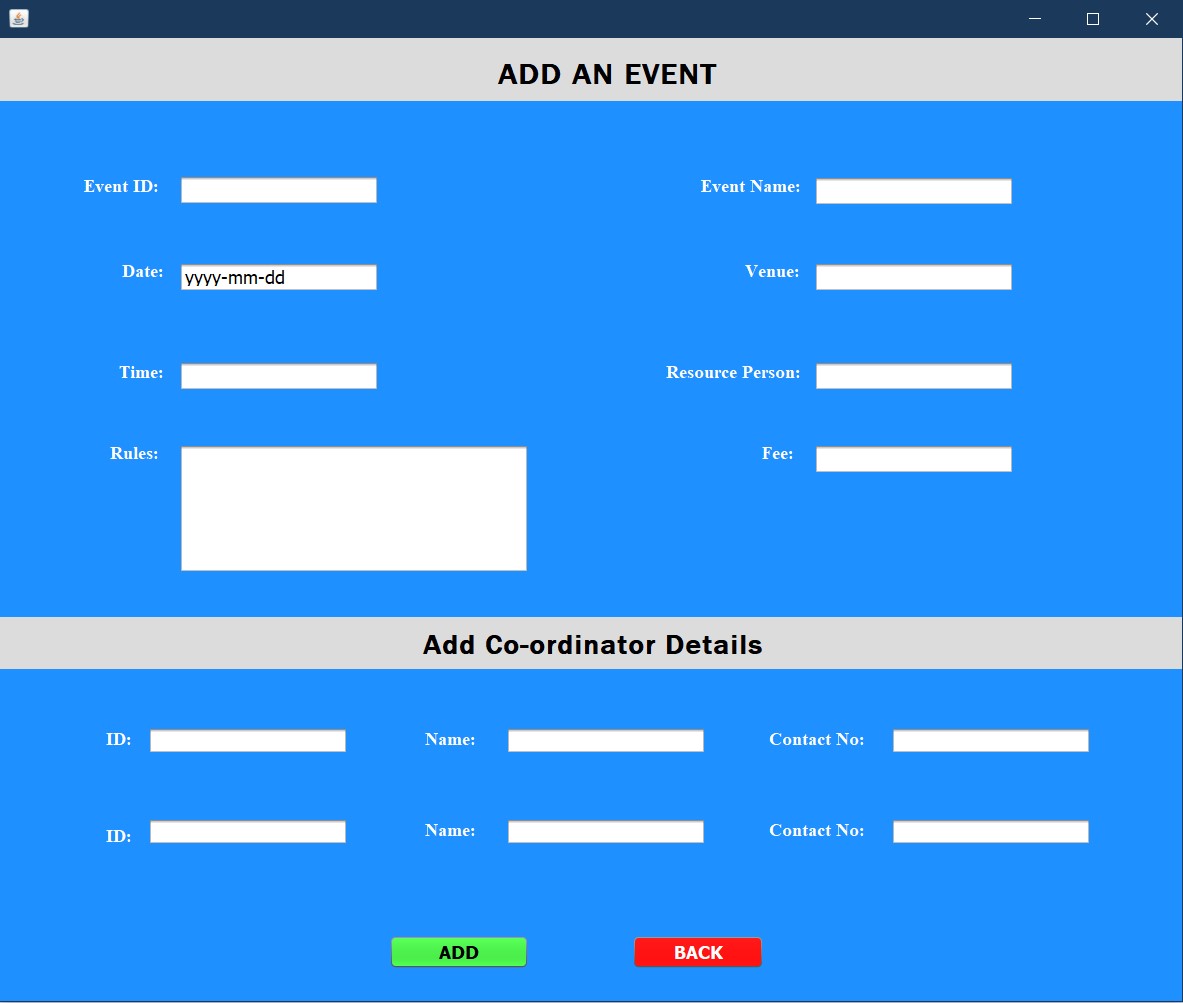
**2. Registration page:**

**3. About page:**

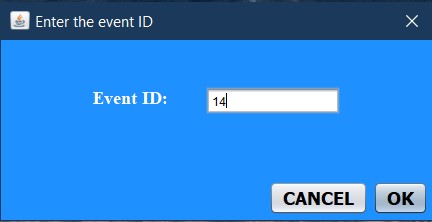
**4. Admin login:**

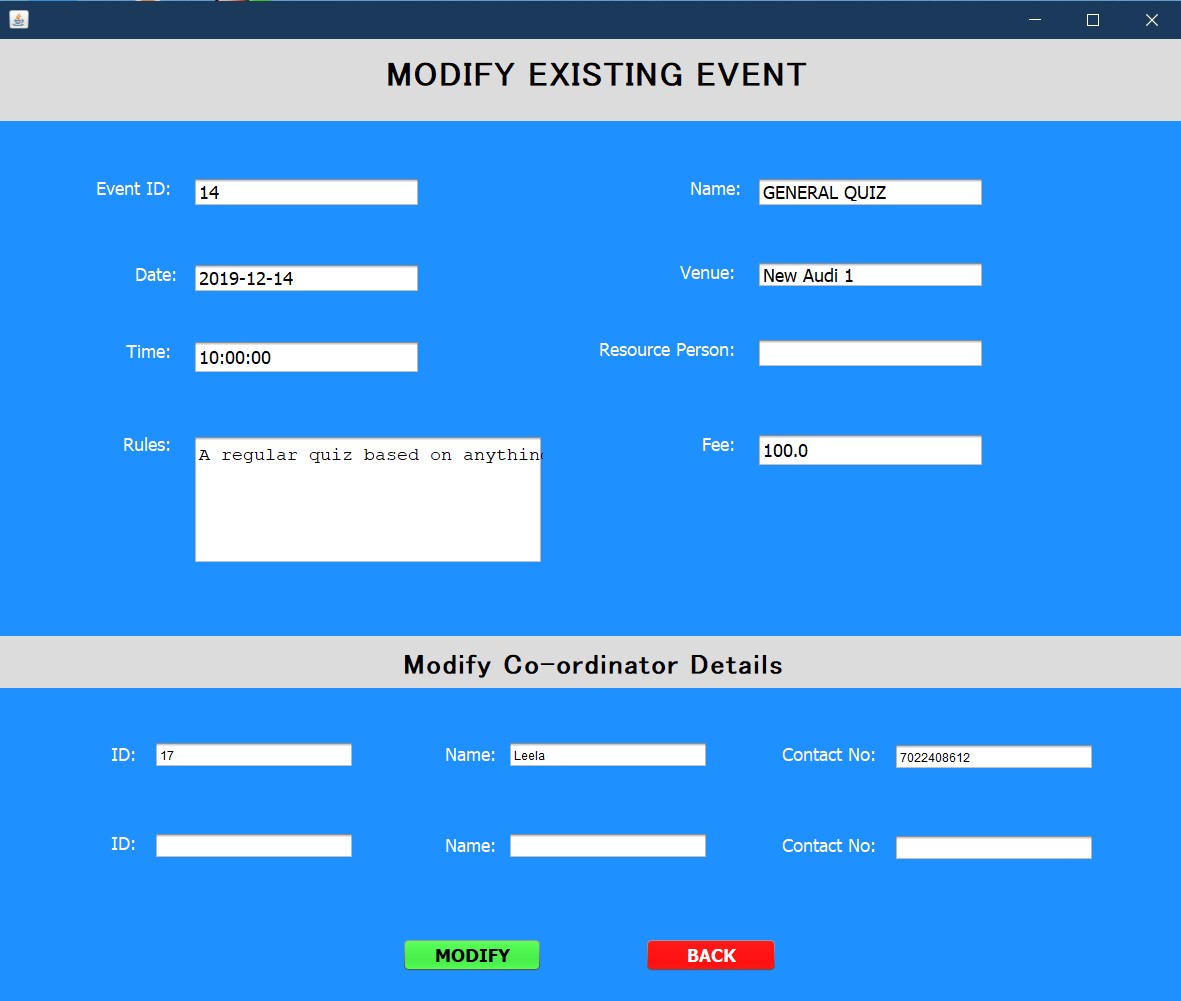


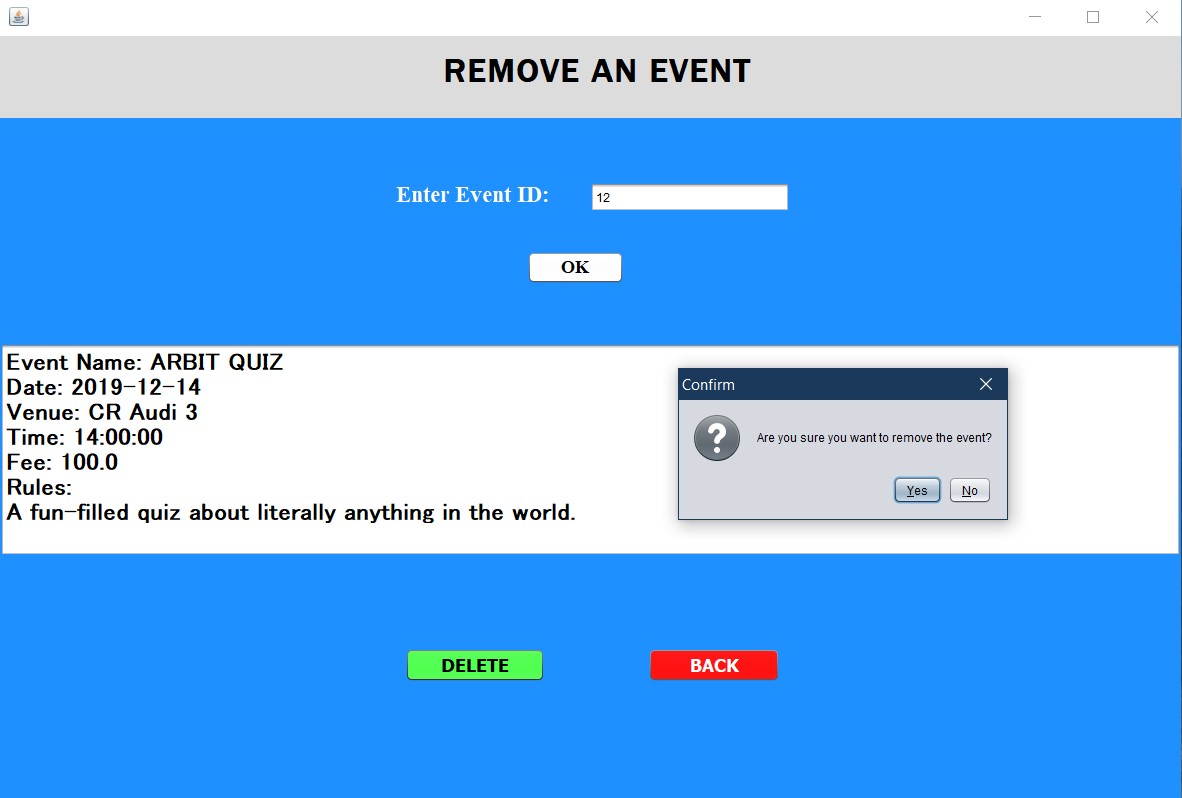
**5. After admin login:**

**6. Add Event page:**

1. **Modify Event page:**





**8. Removing an event:**

**9. View Registrations:**

**10. Updating Registrations of Participants:**

