



GL Bajaj Audi Booking Web-App

Introduction:

The main objective of building this application was to bring in the concept of automation and to remove the paper work involved in the traditional method of booking auditorium through manual work of management of bookings. This idea came into the mind of two young students of the G. L. Bajaj namely **Mukul Kumar** and **Varun Agrawal** and it is their small contribution towards the development of college. This application was made keeping in mind all the features and functions needed to efficiently manage an auditoriums booking.

Features of the system:

➤ User and Admin Login and Dashboard:

Though this application has the ability for user to request for booking but admin has the whole authority and responsibility for approving and disapproving the user's request. This application has given separate set of services to both user and admin and some extra services to administrator keeping in mind their purpose.

➤ Allotment of the auditorium according to the number of attendees:

This application is focused on allotting the auditorium to user in the most effective and ecofriendly way possible, so that it doesn't harm the environment with the excessive usage of present inventions. This allowance of auditorium depending on the number of gathering facility is totally dependent on administrator.

➤ Check previous bookings:

The system keeps track of auditorium status and advance bookings in a well-maintained database. Thus it becomes easier for anyone to search any record and data for their future reference. The users of this application can track all the information of booking dates and requests as per their requirement.

➤ Email Notification:

The system will inform the respective customers through an email about their status of booking request whether approved or disapproved.

➤ Proper Validations:

The system is consisting of all the validations wherever needed and necessary. One such validation is limitation on booking of auditorium on any previous data irrespective of present or future date, or, filling of all the details in the well-set requesting form.

➤ Proper Serving and Tracking of requests:

Since the application is stored on server and its database is also stored on a remote server, therefore it is capable of fulfilling any query at any time with the least possible latency time. Administrators can anytime track any request and approve requests.

➤ User Oriented GUI:

The system is built with easy user navigations and user interfaces. Proper usage of colors and interactivity is added wherever required. Proper Inclusion of photos of auditorium is also added for the reference of user.

Hardware Requirement:

There is no hardware requirement as such for this application except for the computer itself which is required for accessing the internet and requesting for the booking.

Software Requirement:

Operating System: Windows, Linux, MacOS

Language: J2EE

Database: MySQL Server

Browser: Any of Mozilla, Chrome, Safari

Webserver: Tomcat 8

Software Development Kit: Java JDK 1.8

Implementation Methodology:

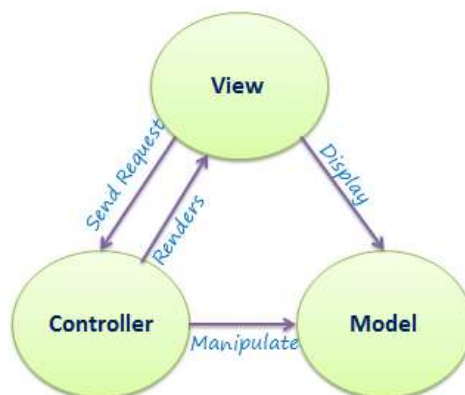
Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

Model - The lowest level of the pattern which is responsible for maintaining data.

View - This is responsible for displaying all or a portion of the data to the user.

Controller - Software Code that controls the interactions between the Model and View.

MVC is popular design pattern and also as it separates the application logic from the user interface layer. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.



Workflow of Application:

- Admin will manage auditorium booking via a booking UI which will allow him or her to observe the availability and unavailability of the auditorium for any date and hourly time interval also.
- Admin will also manage user logins, will have the option to deactivate any user
- An external user has to first register and his or her login will have to be approved by Admin and then can Login.
- Once approved external users can login and raise a request for auditorium booking
- Admin can deactivate any user's login also
- Managing Booking activity will be done via a web service logically at the backend
- An easy to use UI will be provided for checking the availability of auditorium
- Application will have the option to register users on the basis of their departments as this part also needs to be managed (client needs to provide us departments related info so that we can manage them at backend)