# OnVENUE SYSTEM

# Project Report

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

*the degree of Master of Computer Applications of A P J Abdul Kalam*

*Technological University*

**Submitted by:**

**ALPHONS BABY (Reg no:SJC20MCA-2011)**



# MASTER OF COMPUTER APPLICATIONS
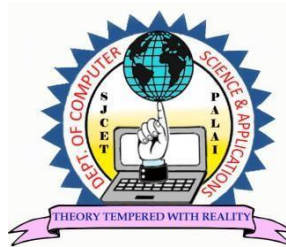
**ST.JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI**

**CHOONDACHERRY P.O, KOTTAYAM**

**KERALA**

**FEBRUARY 2022**

# ST. JOSEPH' S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

(*An ISO 9001: 2008 Certified College*)
**CHOONDACHERRY P.O, KOTTAYAM KERALA**



# *CERTIFICATE*

*This is to certify that the project work entitled **onVENUE SYSTEM** submitted by **Alphons Baby** student of **Third** semester **MCA** at **ST.JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY**, **PALAI** in partial fulfillment for the award of Master of Computer Applications .*

| **Mr. Anish Augustine** | **Dr. Rahul Shajan** | **Mr. Anish Augustine** |
|---|---|---|
| **(Project Guide)** | **(Project Coordinator)** | **(HoD In Charge-MCA)** |

Submitted for the Viva-Voce Examination held on _____

**Examiner 1:**

**Examiner 2:**

# DECLARATION

Me **Alphons Baby**, do hereby declare that the project titled **onVENUE SYSTEM** is a record of work carried out under the guidance Mr. Anish Augustine,Asst.Professor, Department of MCA, SJCET, Palai as per the requirement of the curriculum of Master of Computer Applications Programme of A P J Abdul Kalam Technological University, Thiruvananthapuram. Further, I also declare that this report has not been submitted, full or part thereof, in any University / Institution for the award of any Degree / Diploma.

Place: Choondacherry                            Alphons Baby

Date :                                    (**SJC20MCA-2011**)

# ACKNOWLEDGEMENT

The success of any project depends largely on the encouragement and guidelines of many others. I would like to take this opportunity to express my gratitude to those people who have been instrumental in the successful completion of this project.

First and foremost, I give all glory, honor and praise to **God Almighty** who gave me wisdom and enabled me to complete the project successfully.

I also express sincere thanks, from the bottom of my heart, to my parents for their encouragement and support in all my endeavors and especially in this project.

Words are inadequate to express my deep sense of gratitude to **Dr. J. David,Principal, SJCET, Palai** for allowing me to utilize all the facilities of our college and also for his encouragement.

I extend my sincere gratitude to **Mr. Anish Augustine, HOD Incharge, Department of MCA**, **SJCET, Palai** who has been a constant source of inspiration and without his tremendous help and support this project would not have been materialized.

I owe a particular debt of gratitude to my internal project guide, **Mr. Anish Augustine, Asst. Professor, Department of Computer Science and Applications, SJCET, Palai** for all the necessary help and support that he has extended to me. His valuable suggestions, corrections and the sincere efforts to accomplish my project even under a tight time schedule were crucial in the successful completion of this project.

I extend my sincere thanks to all of our teachers and non-teaching staff of SJCET, Palai for the knowledge they have imparted to me over the last three years.

I would also like to express my appreciation to all my friends for their comments, help and support.

# SYNOPSIS

The venue booking system helps streamlining of reservation and usage of common and individual facilities by college within a facility or a building complex. The facility booking, approval and view across college enhances operational efficiency. Meeting rooms, conference halls, seminar hall, college space auditoriums, training facilities etc. can be booked through the system. The objective of this system, the user can search the facilities and can book from anywhere. As we know there are many facilities available in the field of college. This includes facilities like conference halls, seminar hall, college space auditoriums etc. In early stages, staff used to book these facilities manually. This process was time consuming and included lot of paper works. It has been very hard to book these facilities. Now, staff can book these facilities through online platforms. They don't have to go anywhere to avail these facilities. By checking the availability of facility, the staff can book this facility through online. The staff can book these facilities for a short period of time. The facility booking system does not only assist the facilities unit to achieve higher quality but it also helps to simplify the booking process and raise the efficiency in management.

# TABLE OF CONTENT

# 1.INTRODUCTION

## 1.1 ABOUT PROJECT

The project named "onVENUE" is a software provides that deals with college venue automation. This system provide efficient way of managing the venue booking information. This project is designed with a goal to making the existing system more informative, reliable, fast and easier. This project is based on the college. onVENUE is the easiest way to manage venue booking functionalities of a institute.

The main objective of this project is to mobility and automation to the process of venue booking in an institute. The application will be used by staff, principle. As we know there are many venues available in the field of college. This include facilities like conference halls, seminar hall, college space auditoriums etc. In early stages, staff used to book these venues manually. This process was time consuming and included lot of paper works. It has been very hard to book these venues. Now, staff can book these venues through online platforms. They don't have to go anywhere to avail these venues. By checking the availability of venue, the staff can book this venue through online and book these venues for a short period of time. It also helps to simplify the venue booking process and raise the efficiency in management.

# 2. ABOUT THE ORGANIZATION

## 2.1 ABOUT THE ORGANIZATION

The onVENUE system helps streamline reservation and usage of common and individual facilities by college within a facility or a building complex. Meeting rooms, conference halls, seminar halls, college space auditoriums, training facilities, etc. can be booked through the system and some case admin can reschedule booked venues and also send a notification to the booked staff about the rescheduled details. The objective of this system, the user can search the venues and can book from anywhere. As we know there are many venues available in the field of college. This includes facilities like conference halls, seminar halls, college space auditoriums, etc. In the early stages, staff used to book these facilities manually. This process was time-consuming and included a lot of paper works. It has been very hard to book these facilities. Now, staff can book these facilities through online platforms. They don't have to go anywhere to avail these facilities. By checking the availability of the facility, the staff can book this facility online. The staff can book these venues for a short period of time and it also helps to simplify the booking process and raise the efficiency in management.

# 3. SYSTEM ANALYSIS

System analysis is a structured method for identifying and solving problems. Analysis implies breaking something into its parts so that the whole may be understood. The definition of system analysis not only process analysis but also that of synthesis, which implies the process of putting parts together to form a new whole. The purpose of developing onVENUE is to computerized the traditional way of arranging the venue booking details and it also reduces the paper work, details of all venue list, booked status will be available in website. The main purpose of this project that manage the institution venue booking. The main objective of this project is to mobility and automation to the process of managing venues in an institute.

## 3.1 EXISTING SYSTEM

The existing system is completely manual. The staff can book venues manually. There are lot of paper works and time consuming. The current system is dealing on the manual basis. Staff have to go directly to the office to book the venue in a particular date. It needs huge amount of paper works to maintain the booking details. A huge expenditure and lots of time is spending in the existing system. Tracking and retrieving of data from bulk of paper is a difficult process. The existing system is very paper based in institution. Relevant and irrelevant information of venue booking are entered and stored in the same records, which is very clumsy and process. In this process very difficult to maintain the details of venue booking in a proper way. The venue booking, approve and reschedule procedure is very inefficient. Further, there are chances of data misplacement and wrong data entry. The system is still very inflexible. In the existing system, the manual process, receiving data from staffs are done through manual records. These records are entered in manual process.in this process will take long time.

## 3.2 PROPOSED SYSTEM

The proposed system is an online web application. This system is proposed to control and avoid the limitations of existing system. The goal of proposed system is to increase the efficiency by speeding up the process and bringing down the work load. In the proposed system, the staff need not go anywhere, he can just get all information about venues in a single click. The staff needs to login, so that he can select the venues and book by checking the availability. It is possible for

the staff to access the details regarding the venues from anywhere around the globe. The main objective of the proposed system is to provide a user-friendly interface. The system, which is proposed, now computerizes all the venue booking details that are maintained manually.

## 3.3 SOFTWARE REQUIREMENT SPECIFICATION

The main purpose of this project that manage the institution venue booking. The main objective of this project is to mobility and automation to the process of managing venues in an institute. In early stages, staff used to book these facilities manually. This process was time consuming and included lot of paper works. It has been very hard to book these facilities. Now, staff can book these facilities through online platforms. They don't have to go anywhere to avail these facilities. By checking the availability of facility, the staff can book this facility through online. The staff can book these venues for a short period of time and it also helps to simplify the booking process and raise the efficiency in management.

In today's life, manual booking is not the appropriate way of booking things. It is changes according to our workload and situation. The need of this project is to manage the venues and booking venues easily. It is complete time management system.

This project is a web application that aims introducing a wide variety of technology for managing venues and booking the venues. The feature provided by this application including secure login and staff can search the venues. Also they can book the available venue.

### 3.3.1 SOFTWARE REQUIREMENT

The software requirements specification is a means of translating the ideas in the minds of clients into a formal documentation. This document forms the development and software validation. The basic reason for the difficulty in software requirement specification comes from the fact that there are three interested parties the client, the end users and the software developer. The requirements document has to be such that the client and the user can understand easily and the developers can use it as a basis for software development. Due to the diverse parties involved in software requirement specification, a communication gap exists. This gap arises when the client does not understand software or the software development processor when the developer does not understand the client's problem and application area.

### 3.3.2 HARDWARE REQUIREMENT

Requirements analysis is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. This step acquiring all the facts problem specification such as identifying the desired result determining what information is needed to produce these results and figuring out what process must be carried out to proceed to get the accurate result.

## 3.4 FEASIBILITY ANALYSIS

A feasibility study is carried out to select the best system that meets performance Requirements. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behavior of the system. Feasibility study is a test of system proposed regarding its workability, impact on the organization, ability to meet the needs and effective use of resources. The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as technical, economical and behavioral feasibilities. The proposed system must be evaluated from the technical point of view first and if this technical impact on the organization must accessed. Generally feasibility studies are undertaken within right time constraints. It should be conducted completely ad no fundamental errors of judgments are made. If compatible social and technical systems can be devised then the system must be tested for economic feasibility. It is very important to evaluate the feasibility of a project at the earliest possible time. Feasibility study and risk analysis are related in many ways. If the project risk is great, the feasibility of producing quality software is reduced.

The key factors considered during feasibility study are:

➢ Technical Feasibility
➢  Economic Feasibility
➢ Behavioral Feasibility

### 3.4.1 TECHNICAL FEASIBILITY

It is a study of resources availability that may affect the availability to achieve an acceptable system. It is essential that the process of analysis and definition to conducted in parallel with the assessment of technical feasibility. It centers on the existing computer system and to what extent it can support to the proposed system. This involves the financial considerations to accommodate technical enhancements. If the budgets is a serious of constraint, the project is judged as not feasible. The handling of the proposed system does not Require the changing of the existing configuration of the system.  The technical needs of the system may include: front end and backend selection. An important issue for the development of a project is the selection of suitable front end and backend. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project.

### 3.4.2 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of the system and is commonly known as cost benefit analysis, the procedure made costs. The result of a comparison is found out and changed if needed. This is an ongoing effort that improves the accuracy at each phase of the system life cycle. If a benefit outweighs costs, then decision is made to decide and implement the system. Otherwise, further justification or alternation in the proposed system will have to be made and the process is repeated. It has been proven that the proposed system is economically feasible since it provides several cost benefits. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs.

This project has only one time implementation cost and it is easily available. There for this system is cost effective. So this project is economically feasible as required software is easily available

### 3.4.3 BEHAVIORAL FEASIBILITY

Behavioral feasibility is concerned with the working of the system after its installation. The company has a good record of development, installation and maintenance of systems for its clients. So this system can be installed in the client environment and the company will help in the

maintenance of the system in future. Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project.

The proposed project would be beneficial to all Organizations that, it satisfies the objectives when developed and installed. All the behavioral aspects are considered carefully. Thus the project is behaviorally feasible and it can also be implemented easily

MCA 2020-2022

## 3.5 DATA FLOW DIAGRAM

**Level-1.1**

MCA 2020-2022
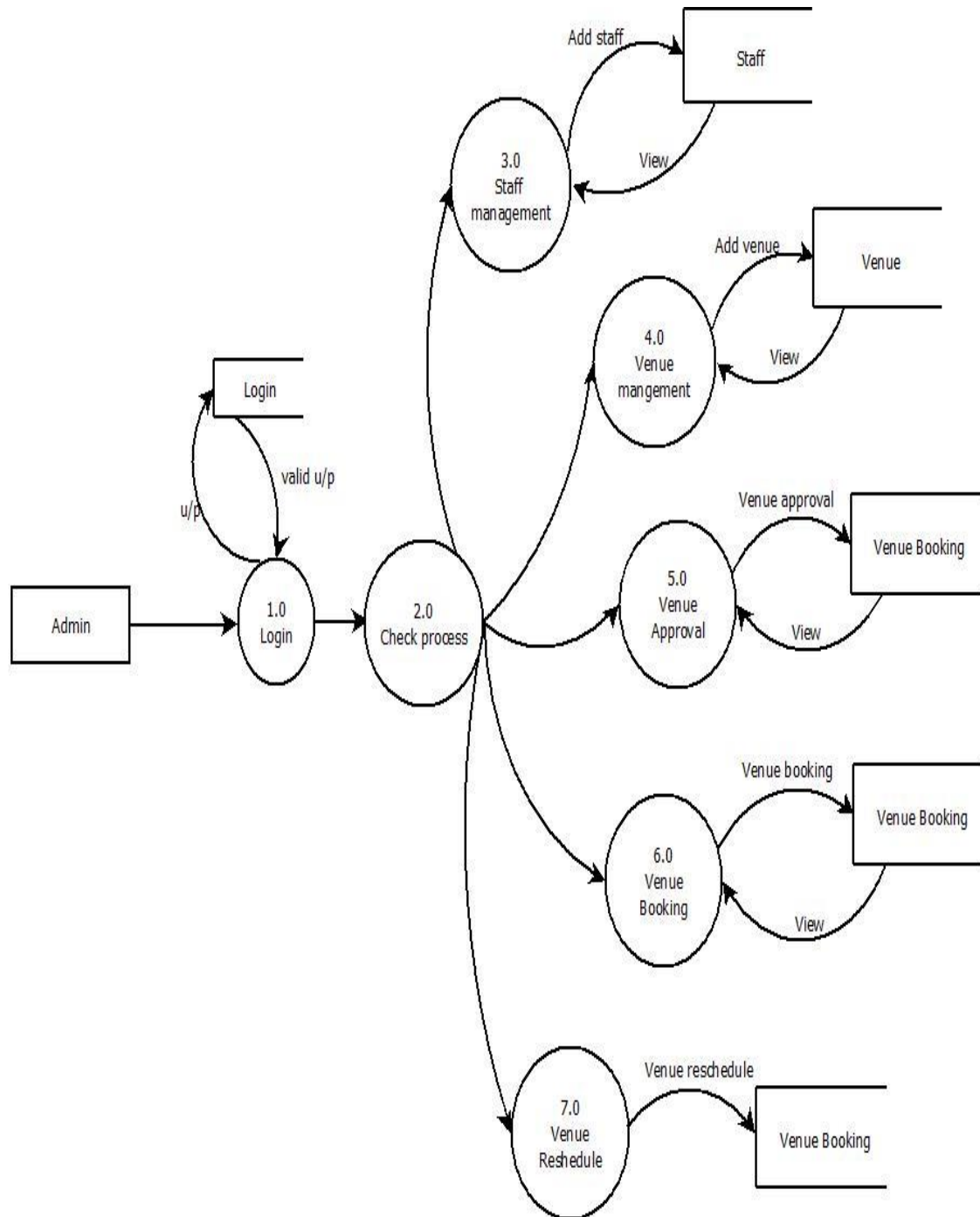
**Level-1.2**
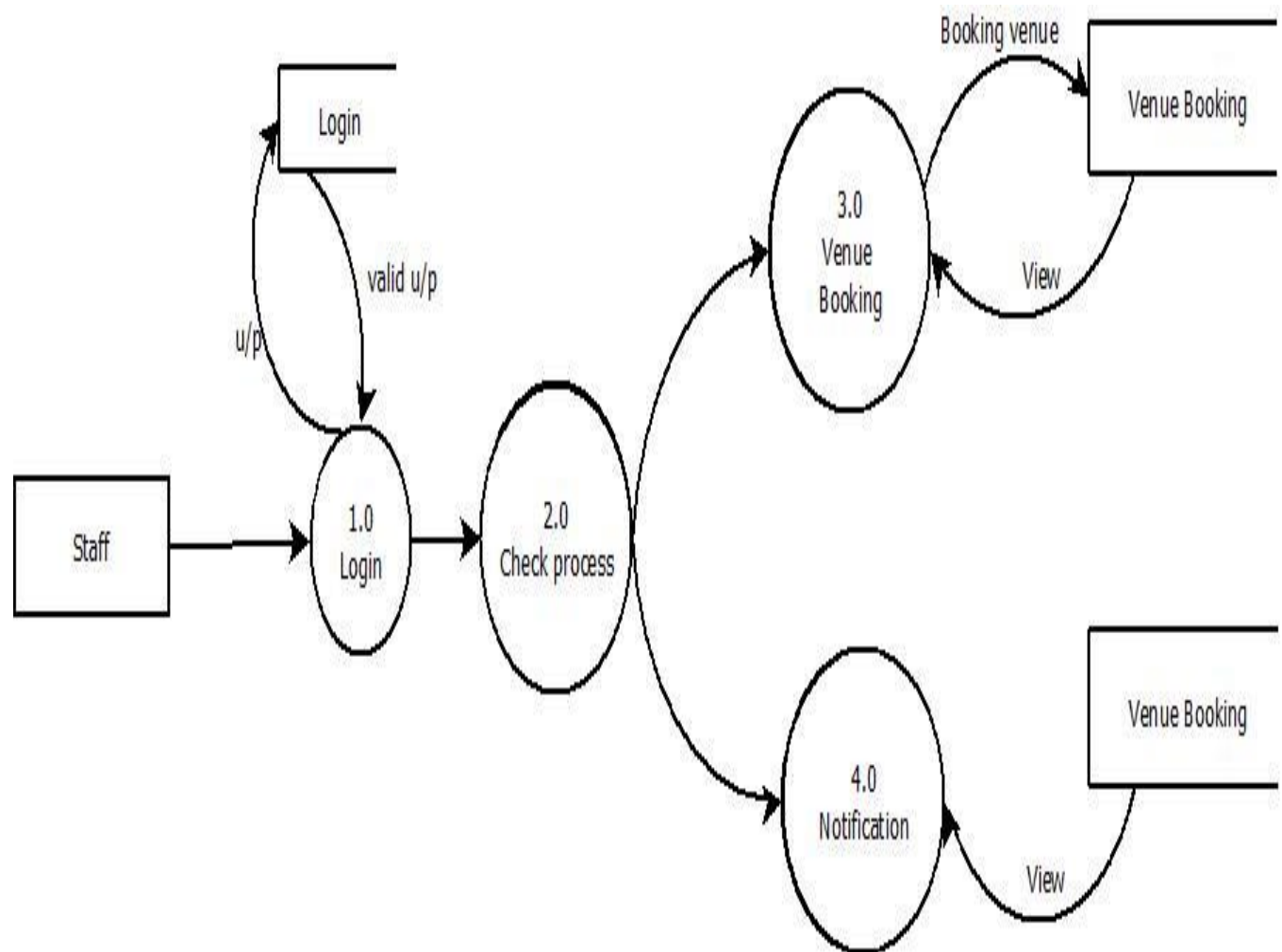
# 4. SYSTEM DESIGN

System designing is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. it is a solution to a "how to" approach compared to system analysis which is a "what is" orientation. it translates the system requirements into ways of making them operational. The design phase focuses on the detailed implementation of the system recommended in the feasibility study.

The system which is in making is developed by working on two different modules and combining them to work as a single unit. That single unit is the one which is known as the new software. we go through the different design strategies to design the system we are talking about. In the input design we decide which type of input screens are going to be used for the system in making. In the output design we decide the output screens and the reports that will be used to give the output and in the database design we decide what all tables will be required and what all fields will be there in those tables. Each of them discussed briefly below

## 4.1 INPUT DESIGN

Input design is one of the most expensive phases of the operation of computerized system and often the major problem of a system. A larger number of problems with a system can usually be traced back to fault input design and methods. Needless to say, therefore that the output data is the block of a system and has to be analyzed and designed with the most consideration

It is the process of converting the user-oriented description of inputs into a computer based business information system to a programmer-oriented specification. The objective of input design is to create an input layout that is easy to follow and prevent operator errors. It covers all phases of input from creation of initial data into actual entry of the data to the system for processing. The input design is the link that ties the system into world of its users

The user interface design is very important for any application. The interface design defines how the software communication within itself, to system that interpreted with it and with human who use it. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format

MCA 2020-2022

and interaction dialogue for giving the right message and help for the user at right time are also considered for the development of the project.

THE SCREENSHOT OF THE INPUT FORMS:

**Form Of Login**



**Staff Registration form**

MCA 2020-2022

**Venue Registration form**



**Venue Booking form**

## 4.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the user and to other systems through outputs. In the output design it is determined how the information is to be displayed for immediate need and also the hard copy output. It is the most important and direct source information to the user. Thus output design generally refers to the result and information that are generated by the system. For many end users, output is the main reason for developing the system and the basis on which they are evaluate the usefulness of application.

The objective of a system finds its shape in terms of the output. The analysis of the objective of the system leads to determination of outputs. Outputs of a system can take various forms. The most common are reports , screens, printed form, graphical drawing etc.

The outputs also vary in terms of their contents, frequency, timing and format. The users of the output, its purpose and sequence of details to be printed are all considered. The output forms a system in the justification for its existence. If the outputs are inadequate in any way, the system itself is inadequate. The basic requirements of output are that it should be accurate, timely and appropriate, in terms of content, medium and layout for its intended purpose. Hence it is necessary to design output so that the objectives of the system are met in the best possible manner. The outputs are in the form of reports

THE SCREENSHOT OF THE OUTPUT FORMS:

**Staff Details form**



onVENUE | Staff ▾ Venue ▾ Booking Status ▾ | Home | Logout

**STAFF DETAILS**

| name | DOB | department name | Qualification | Gender | Address | Edit | Delete |
|------|-----|-----------------|---------------|--------|---------|------|--------|
| Alan sabu | Feb. 23, 2022 | MCA | mba | female | poothottathil | Edit | Delete |
| Alan sabu | Feb. 23, 2022 | MCA | mba | male | poothottathil | Edit | Delete |
| ADMIN | Feb. 1, 2022 | MBA | mba | male | poothottathil | Edit | Delete |

MCA 2020-2022

**Venue Details form**

onVENUE    Staff ▾  Venue ▾  Booking Status ▾                    Home    Logout

## VENUE DETAILS

| Venue name | Location | Description | Seating Capacity | Edit | Delete |
|------------|----------|-------------|------------------|------|--------|
| auditorium | btech department | 10-3:00 pm | 250 | Edit | Delete |

**Venue Approval/reject form**

onVENUE    Staff ▾  Venue ▾  Booking Status ▾                    Home    Logout

## Venue Approval

| Venue name | Event Details | No of Person | Start Date | End Date | | | |
|------------|---------------|--------------|------------|----------|--|--|--|
| auditorium | annual day | 150 | March 1, 2022 | March 1, 2022 | Check | Approval | Reject |

MCA 2020-2022

**Booking status form**

## 4.3 TABLE DESIGN

**TABLE NAME: Login**

| FILED NAME | DATA TYPE | LENGTH | KEY | DESCRIPTION |
|---|---|---|---|---|
| id | int | 50 | Primary key | |
| username | varchar | 50 | Foreign key references to table addstaff | Username for admin and other modules |
| password | varchar | 50 | | Password for admin and other modules |
| Role | varchar | 50 | | Role of admin and other modules |

**TABLE NAME: Staff**

| FILED NAME | DATA TYPE | LENGTH | KEY | DESCRIPTION |
|---|---|---|---|---|
| StaffId | Varchar | 50 | Primary key | Identity number of staff |
| StaffName | Varchar | 50 | | Name of staff |
| Dest | Varchar | 50 | | Destination of staff |
| dept | Varchar | 50 | | Department of staff |
| Course | Varchar | 50 | | Course of staff |
| Qualification | Varchar | 50 | | Qualification of staff |
| dateofbirth | Date | 50 | | Date of birth of staff |

| Gender | Varchar | 50 | | Gender number of staff |
|---|---|---|---|---|
| Address | Varchar | 50 | | Address of staff |
| Photo | file | | | photo of staff |

**TABLE NAME:Venue**

| FILED NAME | DATA TYPE | LENGTH | KEY | DESCRIPTION |
|---|---|---|---|---|
| Venue_id | Varchar | 50 | Primary key | Identity of Venue |
| Venue_name | Varchar | 50 | | Name of Venue |
| Location | Varchar | 50 | | Location of Venue |
| Description | Varchar | 50 | | Description of Venue |
| Seat | Varchar | 50 | | Seating capacity of venue |

**TABLE NAME:Venue Booking**

| FILED NAME | DATA TYPE | LENGTH | KEY | DESCRIPTION |
|---|---|---|---|---|
| Bid | Varchar | 50 | Primary key | Identity of venue booking |
| StaffId | Varchar | 50 | Foreign key references to table Staff | StaffId of staff |
| Venue_id | Varchar | 50 | Foreign key references to table addVenue | id of Venue |

| Eventname | Varchar | | | name of event conducted in the Venue |
|---|---|---|---|---|
| Description | Varchar | 50 | | Approval or rejected of Venue |
| Start date | Date | | | Starting Time of meeting |
| End date | Date | | | Ending Time of meeting |
| No of person | Varchar | 50 | | No of person in the meeting |
| HPL | file | 50 | | HOD permission letter |
| Status | Varchar | 50 | | Coordinator approval |

## 4.4 PROCESS DESIGN

A Process Design Document  is a document that captures the flow of a business process to be developed within RPA. It typically contains the process flow and sequence of steps for the current manual process as well as the automated process, and the various exceptions, conditions and rules of the business process to be automated. Normally authored by a Business Analyst, the PDD is the most common mechanism used in RPA implementations to communicate business processes to be automated and facilitate development.

It's curious that RPA - which is based on making high-volume, routine operations much more efficient and reliable - uses such an outdated and cumbersome means of communication like the PDD. At a time where efficiency is so accessible via digital, centralized tools that can be integrated within any enterprise architecture, surely there must be a better alternative.

MCA 2020-2022

# 5. SYSTEM TESTING & IMPLEMENTATION

It is a systematic technique for the program structure while at the same time conducting test to uncover errors associated with the interfaces. The objective is to take unit tested module and build the program structure that has been dedicated by design. All modules are combined in this testing step. Then the entire program is tested as whole. If a set of errors is encountered correction is difficult because the isolation of cause is complicated by vastness of the entire program.

Implementation includes placing the system into operation and providing the users and operation personnel with the necessary documentation to use and maintain the new system. Implementation includes all those activities that take place to convert from the old system to the new.

## 5.1 SYSTEM TESTING

The onVENUE was tested and found to be working as expected. There is abnormal behavior reported during the testing of program. Testing is a method by which we try reducing the testing efforts and bringing out the maximum output. Testing helps us in knowing whether the logical assumptions that we have taken for the system are correct, and if there are correct we have obtained our goal. We test the system to know the errors, to check the validity of the information, to also group the modules with the aim that we meet the system requirements according to the system needs.

Testing is vital to the success of the system. System testing makes logical assumption that if all the parts of the system are correct, we have achieved the mission successfully. System testing is the stage of implementation that is aimed at assuring that the system works accurately and efficiently before the live operation commences.

The candidate system is subject to a variety of tests. A series of test are performed for the performed system before it ready for user acceptance and testing.

The various tests are:

- ➢ Unit testing
- ➢ Integration testing
- ➢ User Acceptance testing

### 5.1.1 UNIT TESTING

The onVENUE was divided into several units and tested individually. Each unit was found to be working satisfactorily. This testing is carried out during the programming stage itself. In this testing step each module is found to be working satisfactorily as regards to the expected output from the module. Using a method called white box testing which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose and where each module or component of the software is tested individually. In the unit test case we will be testing the separate modules of the software. We will test the component by passing data through it and we will be monitoring data to find the errors. We will be looking for entry and exit conditions of the data. We will make sure that all the components work without any trouble

### 5.1.2 INTEGRATION TESTING

After splitting the program into units, the units were tested together to see the defects between each module and function. It is testing to or modules or function together with the intent of finding interface defects between the modules of or functions. Testing completed at as part of unit or functional testing, and sometimes, becomes its own standalone test phase. On a larger level, integration testing can involve putting together of groups modules and functions with the goal of completing and verifying that the system meets the system requirements.

### 5.1.3 USER ACCEPTANCE TESTING

The onVENUE was tested by a client community to see if the program met the requirements defined the analysis stage. It was found to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in the analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with the users at the time of developing and making changes whenever required.

## 5.2 SYSTEM IMPLEMENTATION

Implementation includes placing the system into operation and providing the users and operation personnel with the necessary documentation to use and maintain the new system. Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementationmaynotguaranteeimprovementintheorganizationusingthenew system, as well as, improper installation will prevent. There are four methods for handling a system conversion.

Parallel approach: The old system is operated with the new system.

Direct cut over method: The old system is replaced with the new system.

Pilot approach: Working version of the system is implemented in one part of the organization based on the feedback, changes are made and the system is installed in the rest of the organization by one of the other methods.

Phase-in-method: Gradually implements the system across all users. We have used the direct cut over method in our implementation.

# 6. SECURITY TECHNOLOGIES & POLICIES

## 6.1 SECURITY TECHNOLOGIES

Any computer-based system that manages sensitive information or causes actions that can improperly harm individuals is a target for improper or illegal penetration. Penetration spans a broad range of activities: hackers who attempt to penetrate systems for sport; disgruntled employees who attempt to penetrate for revenge; dishonest individuals who attempt to penetrate for illicit personal gain.

The proposed system, provide security technology attempts to verify that protection mechanisms built into a system will, in fact, protect it from improper penetration. The system 's security must, of course, be tested for invulnerability from frontal attack, but must also be tested for invulnerability from rear attack.

# 7. MAINTENANCE

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes. This section describes the six software maintenance processes as:

➤ The implementation processes contain software preparation and transition activities, such as the conception and creation of the maintenance plan, the preparation for handling problems identified during development, and the follow-up on product configuration management.

➤ The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance group. The maintenance programmer must analyze each request, confirm it (by reproducing the situation) and check its validity, investigate it and propose a solution, document the request and the solution proposal, and, finally, obtain all the required authorizations to apply the modifications.

➤ The process considering the implementation of the modification itself.

➤ The process acceptance of the modification, by confirming the modified work with the individual who submitted the request in order to make sure the modification provided a solution.

➤ The migration process is exceptional, and is not part of daily maintenance tasks. If the software must be ported to another platform without any change in functionality, this process will be used and a maintenance project team is likely to be assigned to this task.

➤ Finally, the last maintenance process, also an event which does not occur on a daily basis, is the retirement of a piece of software.

# 8. CONCLUSION

In today's fast growing world, manual booking of venues is something that requires a change. Booking things online are the way of new world. The procedures involved in manual booking can now be completed in just a few clicks. My project venues booking management system in a great way by avoiding time wastage and many other unwanted expense. I hope that it will create high level of advancements in the field of College and other fields.

## 8.1 SCOPE FOR FUTURE ENHANCEMENTS

In the future, the system can be developed into a well advanced system for facility booking management. The current web based utility can be incorporated to a mobile based application. The system can be upgraded by including more features like SMS facilities, Email notification etc.
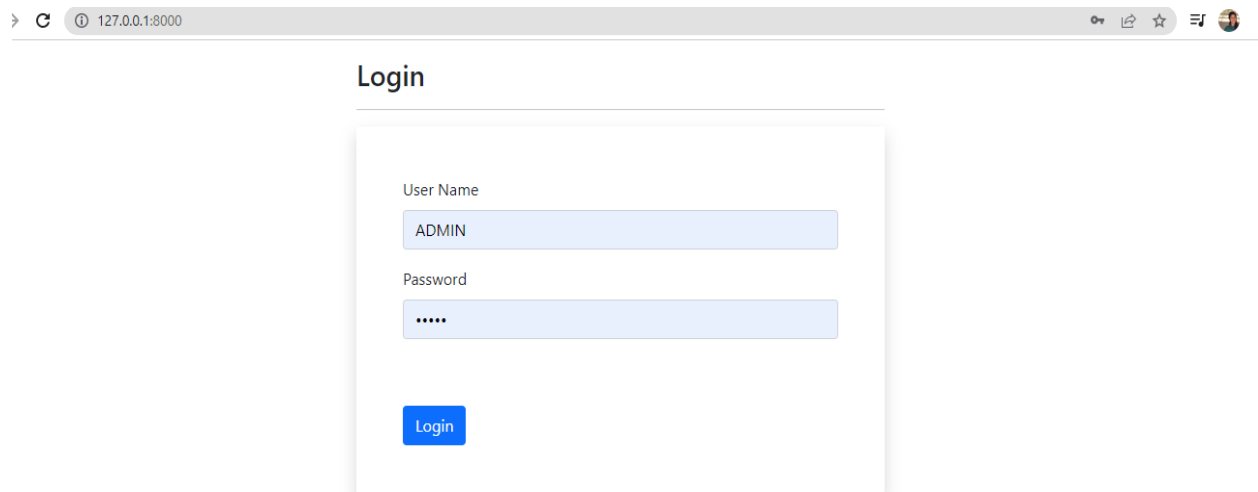
# 9. BIBLIOGRAPHY

- Bootstrap Tutorial, https://www.w3schools.com/bootstrap/
- Django Tutorial :https://www.djangoproject.com/
- Django Tutorial :https ://www.udemy.com/course/python-django-the-practical-guide/learn/lecture
- Udemy course :udemy.com/course/python-django-the-practical-guide/
- Python tutorial :www.codeavail.com/

# 10. APPENDIX

## 10.1 SCREEN SHOTS

Login page



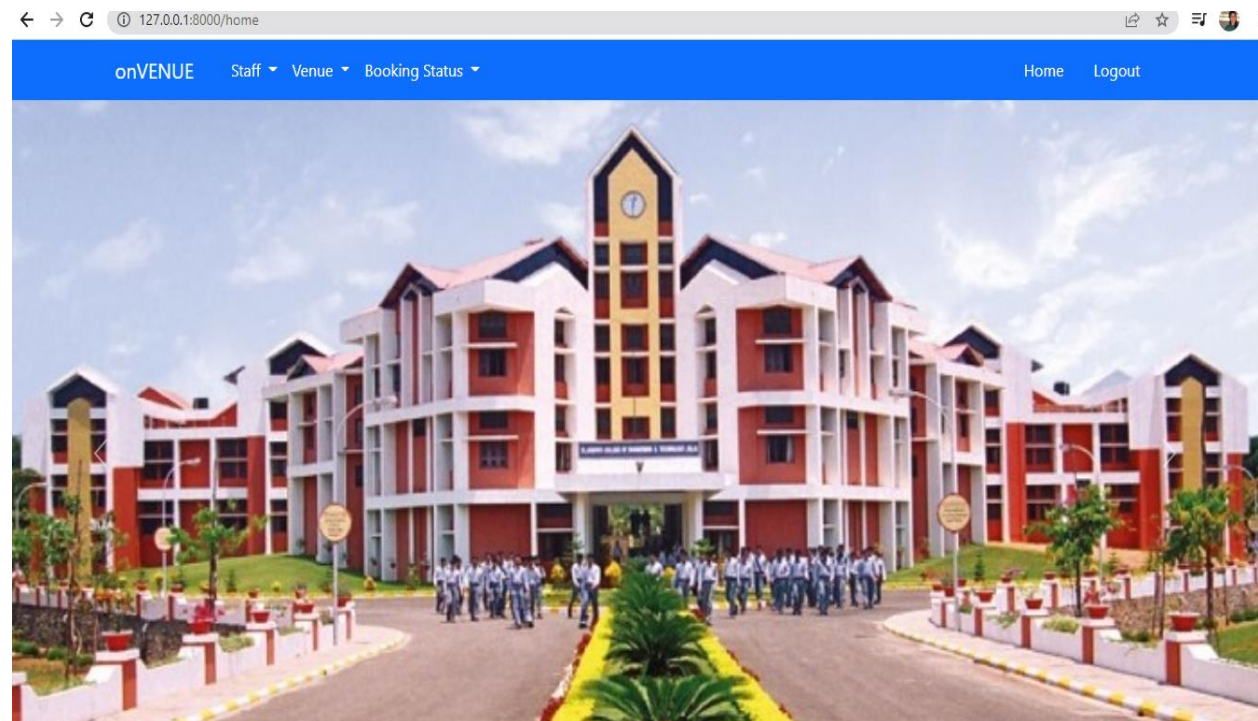Admin home page

MCA 2020-2022

Staff Registration form



Staff Details form

Venue Registration form



Venue Details form

Venue Booking form



Venue Approval/reject form

Booking status form

onVENUE    Staff ▾  Venue ▾  Booking Status ▾                                    Home    Logout

| Booking Search | Enter the Date |
|---|---|
| Booking View | mm/dd/yyyy |

### Venue Booking View

| Venue name | Event Details | No of Person | Start Date | End Date | Description | HPL |
|---|---|---|---|---|---|---|
| auditorium | annual day | 150 | March 1, 2022 | March 1, 2022 | college | |

Venue reschedule form

onVENUE    Staff ▾  Venue ▾  Booking Status ▾                                    Home    Logout

### Venue Reshecdule

| Venue name | Event Details | No of Person | Start Date | End Date | |
|---|---|---|---|---|---|
| auditorium | annual day | 150 | March 1, 2022 | March 1, 2022 | Reshecdule |

MCA 2020-2022

Staff Home page
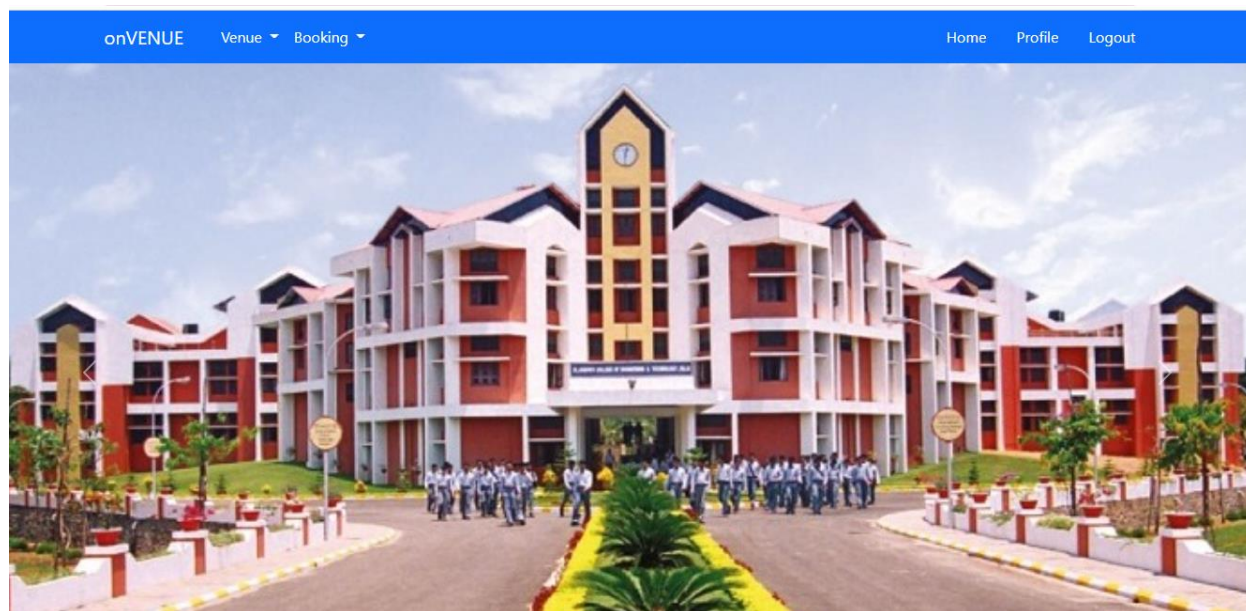


Venue Booking Status



## Venue Booking Status

| Venue name | Event Details | No of Person | Start Date | End Date | Description | Status |
|---|---|---|---|---|---|---|
| auditorium | annual day | 150 | March 1, 2022 | March 1, 2022 | college | |

MCA 2020-2022

## 10.2 REPORTS

Booking status form



Booking status form

## 10.3 CODE

**Views.py**

```python
from django.shortcuts import redirect, render
from django.views import View
from urllib.request import Request
from django.http import HttpResponse, request
from .models import staff,venue,login,venuebook




class adminview(View):
    def get(self,request):

        return render(request,'app/home.html')



def logins(request):
    if  request.method=="POST":
        username = request.POST.get('username')
        password = request.POST.get('password')
        roleadmin='ADMIN'
        l = login(username=username, password=password,role=roleadmin)
        count = login.objects.filter(username=username,
password=password,role=roleadmin).count()
        if count > 0:
            l.save()
            return redirect('home')
        else:
            rolestaff='STAFF'
            l = login(username=username, password=password,role=rolestaff)
            count = login.objects.filter(username=username,
password=password,role=rolestaff).count()
            return redirect('home')
    return render(request,'app/login.html')




def staffregistration(request):
    if request.method=="POST":
        staffname = request.POST.get('staffname')
        dest = request.POST.get('dest')
```

```
        department = request.POST.get('department')
        course = request.POST.get('course')
        Qualification = request.POST.get('Qualification')
        gender = request.POST.get('gender')
        address = request.POST.get('address')
        dob = request.POST.get('dob')

        stud=staff(staffname=staffname,dest=dest,department=department,course=course,
        Qualification=Qualification,gender=gender,address=address,dob=dob)
        stud.save()
        logi=login(username=staffname,password=dob,role=dest)
        logi.save()


    return render(request, 'app/staffregistration.html')


def bookingv(request):
    data=venue.objects.all()
    if request.method=="POST":
        svenue = request.POST.get('svenue')
        eventname = request.POST.get('eventname')
        des = request.POST.get('des')
        nop = request.POST.get('nop')
        sdate = request.POST.get('sdate')
        edate = request.POST.get('edate')
        hpl = request.POST.get('hpl')


v=venuebook(svenue=svenue,eventname=eventname,des=des,nop=nop,sdate=sdate,edate=edate,
hpl=hpl)
        v.save()
    return render(request, 'app/bookingv.html',{'data':data})


def venueb(request):
    if request.method=="POST":
        venuename = request.POST.get('venuename')
        location = request.POST.get('location')
        des = request.POST.get('des')
        seat = request.POST.get('seat')
        v=venue(venuename=venuename,location=location,des=des,seat=seat)
        v.save()
```

```python
    return render(request, 'app/venue.html')

def venuedetails(request):
    data=venue.objects.all()
    return render(request, 'app/venuedetails.html',{'data':data})



def profile(request):
    return render(request, 'app/profile.html')

def address(request):
    return render(request, 'app/address.html')

def orders(request):
    data=venuebook.objects.all()
    return render(request, 'app/orders.html',{'data':data})

def change_password(request):
    return render(request, 'app/changepassword.html')

def mobile(request):
    return render(request, 'app/mobile.html')



def staff_detail(request):
    data=staff.objects.all()
    return render(request, 'app/staffdetail.html',{'data':data})


def venueapproval(request):
    data=venuebook.objects.all()
    return render(request, 'app/venueapproval.html',{'data':data})

def editstaff(request):
    data=staff.objects.all()
    return render(request, 'app/editstaff.html',{'data':data})


def editvenue(request):
    data=venue.objects.all()
    return render(request, 'app/editvenue.html',{'data':data})
def venueres(request):
    data=venuebook.objects.all()
```

```
    return render(request, 'app/venueres.html',{'data':data})
def venuesearch(request):
    data=venuebook.objects.all()
    return render(request, 'app/venuesearch.html',{'data':data})




    #----------------------------staff-----------------

class staffview(View):
    def get(self,request):

        return render(request,'staff/staffhome.html')

def bookingstaff(request):
    data=venue.objects.all()
    if request.method=="POST":
        svenue = request.POST.get('svenue')
        eventname = request.POST.get('eventname')
        des = request.POST.get('des')
        nop = request.POST.get('nop')
        sdate = request.POST.get('sdate')
        edate = request.POST.get('edate')
        hpl = request.POST.get('hpl')


v=venuebook(svenue=svenue,eventname=eventname,des=des,nop=nop,sdate=sdate,edate=edate,
hpl=hpl)
        v.save()
    return render(request, 'staff/booking.html',{'data':data})



def bookingstatus(request):
    data=venuebook.objects.all()
    return render(request, 'staff/bookingstatus.html',{'data':data})

def notification(request):
    data=venuebook.objects.all()
    return render(request, 'staff/notification.html',{'data':data})
def staffvenuesearch(request):
    data=venuebook.objects.all()
    return render(request, 'staff/staffvenuesearch.html',{'data':data})
```

```
def venuestaffview(request):
    data=venue.objects.all()
    return render(request, 'staff/venuestaffview.html',{'data':data})
```

**urls.py**

```
from django.urls import path
from app import views
from django.conf import  settings
from django.conf.urls.static  import static

urlpatterns = [
    #path('', views.home),

    path('', views.logins, name='login'),
    path('home',views.adminview.as_view(),name="home"),
    path('staff-detail/', views.staff_detail, name='staff-detail'),
    path('booking/', views.bookingv, name='bookingv'),
    path('venue/', views.venueb, name='venue'),
    path('profile/', views.profile, name='profile'),
    path('address/', views.address, name='address'),
    path('orders/', views.orders, name='orders'),
    path('changepassword/', views.change_password, name='changepassword'),

    path('registration/', views.staffregistration, name='staffregistration'),
    path('checkout/', views.venueapproval, name='venueapproval'),
    path('editstaff/', views.editstaff, name='editstaff'),
    path('editvenue/', views.editvenue, name='editvenue'),
    path('venuedetails/', views.venuedetails, name='venuedetails'),
    path('venueres/', views.venueres, name='venueres'),
    path('venuesearch/', views.venuesearch, name='venuesearch'),



    # --staff---

    path('homestaff',views.staffview.as_view(),name="homestaff"),
    path('bookingstaff/', views.bookingstaff, name='bookingstaff'),
    path('bookingstatus/', views.bookingstatus, name='bookingstatus'),
    path('notification/', views.notification, name='notification'),
    path('venuestaffview/', views.venuestaffview, name='venuestaffview'),
    path('staffvenuesearch/', views.staffvenuesearch, name='staffvenuesearch'),
```

]

## Models.py

```python
from django.db import models
from django.contrib.auth.models import User

DEP_CHOICE=(("MCA","MCA"),
        ("MBA","MBA"),
        ("MTECH","MTECH"),
        ("BTECH","BTECH")

)
DES_CHOICE=(("STAFF","STAFF"),
        ("INCHARGE","INCHARGE"),
        ("ADMIN","ADMIN")

)
COU_CHOICE=(("MCA","MCA"),
        ("MBA","MBA"),
        ("MTECH","MTECH"),
        ("BTECH-CS","BTECH-CS"),
        ("BTECH-MEC","BTECH-MEC")
)
GEN_CHOICE=(("Male","Male"),
        ("Female","Female")
)
APP_CHOICE=(("APPROVE","APPROVE"),
        ("REJECT","REJECT")
)
ROLE=(("ADMIN","ADMIN"),
        ("STAFF","STAFF")
)


class staff(models.Model):
    staffname=models.CharField(max_length = 50)
    dest=models.CharField(max_length = 50,choices=DES_CHOICE)
    department=models.CharField(max_length = 50,choices=DEP_CHOICE)
    course=models.CharField(max_length = 50,choices=COU_CHOICE)
    Qualification=models.CharField(max_length = 50)
    dob=models.DateField(default="", null=True, blank=True)
    address=models.CharField(max_length = 50)
    gender=models.CharField(max_length = 50,choices=GEN_CHOICE)
```

```
class venue(models.Model):
    venuename=models.CharField(max_length = 50)
    location=models.CharField(max_length = 50)
    des=models.CharField(max_length = 50)
    seat=models.CharField(max_length = 50)

class login(models.Model):
    username=models.CharField(max_length = 50)
    password=models.CharField(max_length = 50)
    role=models.CharField(max_length = 50,choices=ROLE)

class venuebook(models.Model):
    svenue=models.CharField(max_length = 50)
    eventname=models.CharField(max_length = 50)
    des=models.CharField(max_length = 50)
    nop=models.CharField(max_length = 50)
    sdate=models.DateField(default="", null=True, blank=True)
    edate=models.DateField(default="", null=True, blank=True)
    hpl=models.ImageField(upload_to='productimg')


    def __str__(self):
        return str(self)
```