

A PROJECT REPORT ENTITLED  
**“FESTIVA”**

**Submitted to the University of Calicut in Partial fulfillment of  
the Requirement for the Award of Degree Of  
BACHELOR OF SCIENCE IN COMPUTER SCIENCE**



Submitted by  
**MOHAMMED SAHAL A P (POAVSCS009)**  
**VYSHNAV R P (POAVSCS020)**  
**PRABHUL BABU C P (POAVSCS046)**  
**VISHNU P K (POAVSVS048)**

Under the guidance of  
**Mrs. SALEEMA K**  
**(ASSISTANT PROFESSOR)**



**MES PONNANI COLLEGE, PONNANI**  
*(An Institution affiliated to the University of  
Calicut & NACC Re-accredited with A+ Grade)*  
**PONNANI SOUTH P.O, MALAPPURAM-679586**  
**MARCH 2024**

A PROJECT REPORT ENTITLED  
**“FESTIVA”**

**Submitted to the University of Calicut in Partial fulfillment of  
the Requirement for the Award of Degree Of  
BACHELOR OF SCIENCE IN COMPUTER SCIENCE**



Submitted by  
**MOHAMMED SAHAL A P (POAVSCS009)**  
**VYSHNAV R P (POAVSCS020)**  
**PRABHUL BABU C P (POAVSCS046)**  
**VISHNU P K (POAVSVS048)**

Under the guidance of  
**Mrs. SALEEMA K**  
(ASSISTANT PROFESSOR)



**MES PONNANI COLLEGE, PONNANI**  
*(An Institution affiliated to the University of  
Calicut & NACC Re-accredited with A+ Grade)*  
**PONNANI SOUTH P.O, MALAPPURAM-679586**  
**MARCH 2024**



**MES PONNANI COLLEGE, PONNANI**  
*(An Institution affiliated to the University of  
 Calicut & NACC Re-accredited with A+ Grade)*  
**PONNANI SOUTH P.O, MALAPPURAM-679586**

## **DEPARTMENT OF COMPUTER SCIENCE**

# **CERTIFICATE**

This is to certify that the project work entitled “**FESTIVA**” is a bonafide record of work done by **MOHAMMED SAHAL A P, PRABHUL BABU C P, VYSHNAV R P, VISHNU P K** submitted on partial fulfillment of the requirements for the award of degree of **BSc. COMPUTER SCIENCE** of the Calicut University under my supervision.

Date:

Signature of the guide

Counter Signed by H O D

Eternal viva-voce conducted on \_\_\_\_\_

INTERNAL EXAMINER

EXTERNAL EXAMINER

# **DECLARATION**

**We hereby declare that the project entitled “FESTIVA” submitted to the Calicut University in partial fulfillment of the requirement for the award of degree of BSc. COMPUTER SCIENCE is a record of original work done by us during our period of study at MES PONNANI COLLEGE under the supervision and guidance of Mrs. SALEEMA K, Assistant professor of Computer Science.**

**Place: PONNANI**

**Signature of Candidate**

**Date:**

**MOHAMMED SAHAL A P**

**PRABHUL BABU C P**

**VYSHNAV R P**

**VISHNU P K**

## **ACKNOWLEDGEMENT**

The success of the project depends upon the effort invested. At this pleasure moment of having successfully completed our project. It's our duty to acknowledge and thanks the individuals who have contributed in the successful completion of the project.

We wish to express our heartfelt gratitude to **Dr. ANAS E, Principal, MES PONNANI COLLEGE** for his encouragement and inspiring guidance throughout the preparation of the project.

We express our deep sense of gratitude and sincere thanks to head of the department **Miss. NUSRATH. K** for the valuable guidance to do the project successfully.

We would like to express our special thanks of gratitude to our teachers "**Mrs. Aneeshya C A , Mrs. Saleema K, Mr. Ramees Ali P** " for their valuable guidance and support.

We wish to express our love and respect to our parents, for their support, contribution and encouragement which helped us a lot to complete the project successfully.

We very much thankful to our friends for their support and contribution to complete this project successfully.

## **ABSTRACT**

Efficiently managing program details and the seamless publishing of results are pivotal aspects of effective college administration. The project “Festiva” is meticulously crafted to address these crucial components of academic institution management. Festiva comprises four dedicated modules: Admin, HOD, Staff, and Student, each meticulously designed to simplify the intricate web of program-related tasks. Administrators wield the power to maintain comprehensive program details, exercise control over user access, and fine-tune permissions through the Admin Module. Meanwhile, Heads of Departments (HODs) find their role streamlined with the ability to deftly assign programs to staff members, closely monitor program progress, and assess faculty performance via a user-friendly interface in the HOD Module. Faculty and administrative staff benefit immensely from the Staff Module, which equips them with the tools to efficiently handle volunteering and effortlessly input results. On the student front, the Student Module empowers them with easy access to program information, smooth enrollment, the ability to closely track performance reports.

## **TABLE OF CONTENTS**

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>08</b>
1.1.	PROBLEM DEFINITION	
1.2.	SCOPE OF THE PROJECT	
1.3.	MODULE DESCRIPTION	
<b>2.</b>	<b>SYSTEM STUDY AND ANALYSIS.....</b>	<b>12</b>
2.1.	PRELIMINARY INVESTIGATION	
2.2.	EXISTING SYSTEM	
2.3.	PROPOSED SYSTEM	
2.4.	FEASIBILITY STUDY	
2.5.	SYSTEM SPECIFICATIN	
2.6.	SOFTWARE TOOLS USED	
2.7.	DATABASE SERVICE	
<b>3.</b>	<b>SYSTEM DESIGN.....</b>	<b>27</b>
3.1.	DATA FLOW DIAGRAM	
3.2.	ER DIAGTAM	
3.3.	DATABASE DESIGN	
3.4.	TABLE DESIGN	
3.5.	USER INTERFACE DESIGN	
3.6.	INPUT DESIGN	
3.7.	OUTPUT DESIGN	
<b>4.</b>	<b>SYSTEM TESTING AND IMPLIMENTATION.....</b>	<b>46</b>
4.1.	SYSTEM TESTING	
4.2.	TESTING METHEDOLOGIES	
4.3.	SYSTEM IMPLIMENTATION	
4.4.	SYSTEM MAINTANANCE	
<b>5.</b>	<b>FUTURE ENHANCEMENT... ..</b>	<b>53</b>

<b>6.</b>	<b>CONCLUSION...</b>	<b>55</b>
<b>7.</b>	<b>APPENDIX...</b>	<b>57</b>
7.1.	APPENDIX I:SCREENSHOTS	
7.2.	APPENDIX II:SOURCE CODE	



## INTRODUCTION

## 1. INTRODUCTION

Efficient management of program details and seamless dissemination of results are vital components of successful college administration. In response to this need, the project "Festiva" has been meticulously developed to address these critical aspects of academic institution management. Festiva offers a comprehensive solution, comprising four distinct modules tailored to the specific needs of administrators, Heads of Departments (HODs), faculty and staff, and students. Each module is thoughtfully designed to simplify and streamline the various tasks associated with program management.

Administrators are empowered by the Admin Module, which enables them to maintain detailed program information, manage user access, and fine-tune permissions according to organizational requirements. HODs benefit from the dedicated module designed to facilitate the assignment of programs to staff members, monitor program progress, and evaluate faculty performance through an intuitive interface.

The Staff Module provides faculty and administrative staff with essential tools to efficiently manage volunteering activities and input results, ensuring smooth operation of program-related tasks. Meanwhile, students are equipped with the Student Module, granting them easy access to program information, seamless enrollment processes, and the ability to closely monitor their performance reports.

Through Festiva, academic institutions can enhance their operational efficiency, improve program management processes, and ultimately foster a conducive environment for academic excellence.

### 1.1. PROBLEM DEFINITION

the lack of a comprehensive program management solution presents a significant challenge. Fragmented information management across various platforms and departments leads to confusion and inefficiency, while the absence of dedicated tools for Heads of Departments (HODs) results in difficulties in program assignment and faculty evaluation. Administrators face tedious tasks managing program details and user access permissions, and students encounter barriers to accessing program information and monitoring their performance. Existing systems suffer from integration and usability issues, hampering overall effectiveness. Addressing these challenges requires a unified solution that streamlines administration, enhances communication, and improves user experience across all stakeholders.

### 1.2. SCOPE OF THE PROJECT

Festiva project focuses on developing a basic program management application tailored for academic institutions. This involves creating functionalities for program details management, user access control, progress tracking, and result input. The project aims to address common challenges in program management by providing a user-friendly interface and essential features for administrators, Heads of Departments (HODs), faculty, staff, and students. Through the integration of basic functionalities and usability enhancements, Festiva aims to streamline program management processes and improve efficiency within academic institutions.

### 1.3. MODULE DESCRIPTION

The project contain mainly 4 modules

1. Admin
2. HOD
3. Staff
4. Student

#### **Admin**

- Login
- Verify Student
- Add & Manage HOD
- Add & Manage staff
- View program
- Add and manage category
- Add and manage department
- Assign category

**HOD**

- Login
- Add program and view status
- Assign verified program to staff
- View result

**student**

- Register
- Login
- View and edit profile
- View program and apply
- View assigned program
- View status
- View program result

**Staff**

- Login
- View student
- View assigned program
- View program request and verify
- Upload result
- Assign program to student to coordinate

## **SYSTEM STUDY AND ANALYSIS**

## **2. SYSTEM STUDY AND ANALYSIS**

### **2.1. PRELIMINARY INVESTIGATION**

System study is done in order to understand the problem and emphasize what is needed from system. The information requirements of the user for their competitive world requires such system. The various techniques used in this phase are Observations, Interviews and Discussions. A complete understanding of software requirements is essential to the success of a software development effort. System Analysis refers to an orderly structured process for identifying and solving problems using computer.

It is the most essential part of the project development. It is the process of the gathering and interpreting facts, diagnosing problems and using the information to recommend improvements to the system. Training, experience and common sense are required for the collection of the information needed to do the analysis

### **2.2. EXISTING SYSTEM**

The existing system lacks a centralized platform for efficient management of program details. Administrators often rely on manual methods for maintaining program information, managing user access, and monitoring progress. Heads of Departments face challenges in assigning programs to staff members and evaluating faculty performance. Faculty and administrative staff struggle with managing volunteering activities and inputting results efficiently. Students encounter difficulties accessing program information, and monitoring their performance reports. Overall, the existing system is fragmented, inefficient, and lacks the necessary tools to streamline program management processes.

### **2.3. PROPOSED SYSTEM**

The proposed system, Festiva, addresses the shortcomings of the existing fragmented system by introducing a centralized platform for efficient program management. Festiva's Admin Module enables administrators to maintain program details and manage user access seamlessly. Department Heads can easily assign programs and evaluate faculty performance, while faculty and staff benefit from streamlined volunteering activities and result input. Students gain easy access to program information and performance reports. Festiva promises to enhance operational efficiency and foster academic excellence by streamlining program management processes within academic institutions.

### **2.4 FEASIBILITY STUDY**

A feasibility study is a preliminary study undertaken to determine and document a project's viability. The results of this study are used to make a decision whether to proceed with the project. If it indeed leads to a project being approved, it will - before the real work of the proposed project starts - be used to ascertain the likelihood of the project's success. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative. It, for example, can decide whether an order processing be carried out by a new system more efficiently than the previous one. The feasibility study proposes one or more conceptual solutions to the problem set for the project. The conceptual solution gives an idea of what the new system will look like. They define what will be done on the computer and what will remain manual. It also indicates what input will be needed by the system and what

outputs will be produced. These solutions should be proven feasible and a preferred solution is accepted.

The feasibility study environment enables all alternatives to be discussed and

evaluated. This phase starts with an identification of the main characteristics of the required system. During this stage it is important to collect information as much as possible about the software package that might meet the specification from as many sources as possible.

Normally, the central endeavor of a feasibility study is a cost benefit analysis of various alternatives. It can be defined as a systematic comparison between the cost of carrying out a service or activity and the value of that service or activity. The main benefits are qualitative than quantitative.

A feasibility study could be used to test a new working system, which could be used because:

- The current system may no longer suit its purpose,
- Technological advancement may have rendered the current system obsolete,
- The business is expanding, allowing it to cope with extra work load,
- Customers are complaining about the speed and quality of work the business provides.
- Competitors are now winning a big enough market share due to an effective integration of a computerized system.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration.

Facts considered in the feasibility analysis were

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility
- Behavioral Feasibility
- Legal Feasibility
- Scheduled Feasibility



## **TECHNICAL FEASIBILITY**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on an outline design of system requirements in terms of Input, Output, Fields, Programs, and Procedures. This can be qualified in terms of volumes of data, trends, frequency of updating etc. in order to give an introduction to the technical system.

The system requires normal configuration computer system that are commonly available. The software requirements are Python and Android, Windows 8 or higher versions of OS. Thus proposed system is technically feasible.

## **OPERATIONAL FEASIBILITY**

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and there for it will accept a broad audience.

The proposed system offers:

- Greater user friendliness
- Better output which can be easily interpreted.
- Higher speed.
- Meet the requirements of the organizations.

## **ECONOMIC FEASIBILITY**

This involves questions such as whether the firm can afford to build the system, whether its benefits should substantially exceed its costs, and whether the project has higher priority and profits than other projects that might use the same resources. This also includes whether the project is in the condition to fulfill all the eligibility criteria and the responsibility of both sides in case there are two parties involved in performing any project.

This study presents tangible and intangible benefits from the project by comparing the developments and operational costs. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve the quality of service.

Thus feasibility study should center along the following points:

- Improvement resulting over the existing method in terms of accuracy, timeliness.
- Cost comparison.
- Estimate on the life expectancy of the hardware.
- Overall objective.

## **BEHAVIOURAL FEASIBILITY**

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept broad audience

## **LEGAL FEASIBILITY**

Determines, whether the proposed system conflicts with legal requirements. E.g. a dataprocessing system must comply with the local Data Protection Acts.

## **SCHEDULE FEASIBILITY**

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period.

Schedule feasibility is a measure of how reasonable the project timetable is . given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. You need to determine whether the deadlines are mandatory or desirable.

## **2.5 SYSTEM SPECIFICATION**

The system specification refers to a detailed functional and non-functional description of a system. This term can also be defined as an explicit set of requirement that need to be satisfied be specific system. System specification includes software and hardware specification of project

### **HARDWARE SPECIFICATION**

The selection of hardware is very important in the existence and proper working of any of the software. When selecting hardware, the size and capacity requirements are also

important. The hardware must suit all application developments.

- Processor : i3 or Above.
- System Bus : 32Bit or 64Bit
- RAM : 4 GB or Above
- HDD : 320GB or Above
- Monitor : SVGA colour
- Key Board : Any type of keyboard
- Mouse : Any Type of mouse
- Mobile : Android supported mobile phone

## SOFTWARE SPECIFICATION

One of the most difficult tasks is selecting software, once the system requirement is found then we have to determine whether a particular software package fits for those system requirements. This section summarizes the application requirement.

- Operating System : Windows 7 or above
- Front End : HTML, CSS, Dart
- Back End : Python(Django), DART(Flutter)
- Data Base : MySQL
- Browser : Chrome, Edge, etc..
- IDE : VS Code or Android Studio, Python 3.6 or above, PyCharm

## 2.6 SOFTWARE TOOLS USED

### DJANGO

Django is a high-level Python web framework renowned for its efficiency and versatility in web application development. Developed by experienced developers, Django simplifies complex web development tasks by providing a robust toolkit and follows the Model-View-Template (MVT) architectural pattern, promoting clean and maintainable code.

**Key Features:****Model-View-Template (MVT) Architecture:**

Django's MVT architecture separates the data model, user interface, and application logic, ensuring code organization and scalability. This separation of concerns facilitates code maintenance and enables collaborative development.

**Object-Relational Mapping (ORM):**

Django's ORM seamlessly integrates database operations with Python code, allowing developers to interact with databases using high-level Python objects. This abstraction simplifies database management and reduces the need for raw SQL queries.

**Admin Interface:**

Django's built-in admin interface provides a user-friendly dashboard for managing application data, including CRUD operations on database models. Administrators can effortlessly perform tasks such as adding, editing, and deleting records, enhancing productivity and easing administrative burden.

**Authentication and Authorization:**

Django offers robust authentication and authorization mechanisms out of the box, allowing developers to secure application resources and manage user access permissions effectively. This ensures data integrity and protects sensitive information from unauthorized access.

**URL Routing and Views:**

Django's URL routing system maps URLs to views, enabling developers to define how HTTP requests are handled and responses are generated. Views encapsulate application logic and interact with models and templates to render dynamic content, facilitating flexible and customizable web applications.

**Template Engine:**

Django's template engine empowers developers to create dynamic web pages with minimal code duplication. Templates support inheritance, template tags, and filters, facilitating

code reuse and enhancing maintainability.

**Scalability and Performance:**

Django's scalability and performance optimizations, including caching, database query optimization, and asynchronous request handling, ensure efficient application operation even under heavy load. This scalability makes Django suitable for building high-traffic web applications.

**Community and Ecosystem:**

Django boasts a vibrant community of developers, contributors, and enthusiasts, providing extensive support through documentation, forums, and third-party packages. The Django ecosystem offers a plethora of reusable components, plugins, and libraries, accelerating development and enriching application functionality.

Django's comprehensive feature set, coupled with its emphasis on security, scalability, and maintainability, makes it an ideal choice for developing a wide range of web applications, from simple blogs to complex enterprise systems.

**FLUTTER**

Flutter is an open-source mobile application development framework created by Google. It is used to develop applications for Android and IOS, as well as being the primary method of creating applications for Google Fuchsia. Flutter allows for the creation of natively compiled applications for mobile, web, and desktop from a single codebase 12. Flutter is often used with Dart, which is an object-oriented programming language also developed by Google.

**Key Features:****Single Codebase:**

Flutter allows developers to write code once and deploy it on multiple platforms, such as IOS, Android, web, and desktop. This "write once, run anywhere" approach streamlines development and maintenance.

**Widgets:**

Flutter uses a reactive framework that revolves around widgets – UI components for building the user interface. The extensive widget library enables the creation of complex and customizable interfaces.

**Hot Reload:**

One of Flutter's standout features is the Hot Reload capability, allowing developers to instantly view changes made to the code without restarting the entire application. This accelerates the development process and enhances productivity.

**Expressive UI:**

Flutter provides a rich set of customizable widgets and a flexible design system, empowering developers to create expressive and visually appealing user interfaces.

**Performance:**

Flutter compiles to native ARM code for optimal performance on both iOS and Android platforms. Its high-performance graphics engine, Skia, contributes to smooth animations and fast rendering.

**Community and Ecosystem:**

With a growing community and a wide range of packages and plugins, Flutter offers extensive support for various functionalities, enhancing the development process by providing pre-built solutions.

**Dart language**

Flutter apps are written in the Dart language and make use of many of the language's more Advanced features.

While writing and debugging an application, Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. This allows for fast compilation times as well as "hot reload" with which modifications to source files can be injected into a running application. Flutter extends this further with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

For better performance, release versions of Flutter apps on all platforms use ahead-of-time (AOT) compilation, except for on the Web where code is transpiled to JavaScript.

Flutter inherits Dart's Pub package manager and software repository, which allows users to publish and use custom packages as well as Flutter-specific plugins.

Dart is a modern, open-source, general-purpose programming language developed by Google. It was designed with the goal of being easy to learn, productive, and suitable for a wide range of applications, from client-side web development to server-side and mobile app development.

### **Key Features:**

#### **Object-Oriented:**

Dart is an object-oriented programming language, meaning it revolves around the concept of objects, encapsulation, and inheritance. This promotes code organization and reusability.

#### **Strongly Typed:**

Dart is a statically-typed language, which means variable types are known at compile time. This can help catch errors early in the development process.

Just-in-Time (JIT) and Ahead-of-Time (AOT) Compilation:

Dart uses a combination of JIT and AOT compilation. During development, JIT compilation allows for features like Hot Reload in Flutter. For production, AOT compilation results in optimized, standalone native machine code.

#### **Concurrency Support:**

Dart supports asynchronous programming through its Future and Stream classes. This is especially beneficial for developing responsive and efficient applications, such as in web development.

#### **Dart SDK and Dart Pad:**

Dart comes with a Software Development Kit (SDK) that includes a rich set of libraries and tools. Dart Pad, an online platform, enables developers to experiment with Dart code in a browser environment.



**Cross-Platform Development:**

Dart gained significant attention as the primary language for Flutter, a UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase.

**Garbage Collection:**

Dart features automatic memory management through garbage collection, reducing the burden on developers to manage memory manually.

**Growing Ecosystem:**

Dart has a growing ecosystem with packages and libraries available through the pub.dev repository. This facilitates code reuse and accelerates development.

Dart's versatility, combined with its use in frameworks like Flutter, makes it a compelling choice for developers aiming to build robust and cross-platform applications with a focus on user interfaces.

**2.7 DATABASE SERVICES**

A data base is a collection of inter related data stored with minimum redundancy to serve many quickly and efficiently. The general way is to make information accessing easy, quick, inexpensive and flexible for the user. In data base design several objectives are considered controlling redundancy, ease of learning and use, data dependence, more information at low cost, accuracy and integrity are some of them.

**O RDBMS**

RDBMS is a type of database management system that stores data in the form of related tables. Relational database are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database canbe viewed in many different ways. An important feature of relational systems is that a single database can be spread across several tables. This differs from flat-file database, in which each database is self- contained in a single table.

**O MySQL**

MySQL is an open source relational database and it includes advanced data types. MySQL operates using client/server architecture in which the server runs on the machine containing the database and client connect to the server over the network. MySQL run on all platforms supported by MySQL and provides the most direct means of interacting with the server, so it's the logical client to begin with.

- You need to have the MySQL software installed.
- You need a MySQL account so that you can connect to the server.
- You need a database to work with.

The required software includes the MySQL clients and a MySQL server. The client program must be located on the machine where you will be working. The server can be located on our machine although that is not required. As long as you have permission to connect to it the server can be located anywhere. In addition to the MySQL software you will need a MySQL account so that the server will allow you to connect and create a sample database and its table.

Microsoft SQL Server 2008 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. In this article, we'll cover six of the more frequently used tools: Enterprise Manager, Query analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online. Let's take a brief look at each:

**Enterprise Manager** is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

**Query Analyzer** offers a quick method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

**SQL Profiler** provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system “traces” that log various activities. It’s a great tool for optimizing databases with performance issues or troubleshooting particular problems.

**Service Manager** is used to control the MS SQL Server (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQLServer– Agent processes. An icon for this service Manager to start, stop or pause any one of these services.

**Data Transformation Services (DTS)** provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the “Import and Export Data” wizard found in the SQL Server program group.

## **SYSTEM DESIGN**

### **3.SYSTEM DESIGN**

The most creative and challenging phase of the system life cycle is system design . The term design describes a final system and the process by which it is developed. It refers to the technical specification that will be applied in implementing the candidate system. It also include the construction of the program and the program testing. the key question involved here is “ how the problem should be solved” .

System design is a solution for the question of how to the approach to the creation of anew system. This important phase is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended feasible study. Emphasis is on translating the performance requirements into design specifications. Design goes through logical and physical system; prepare input and output specification; make credit, security and control specification ; details the implementation plan;prepare a logical design walk. Physical design maps out the physical system , plans thesystem implements, devices a test and implementation plan and specifies any new hardware and software.

The first most is to determine how the output is to be produced and in what format. Samples of output and input are presented. Second , input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing , including a list of programs needed to meet the system’s objectives and complete documentation. Finally, details related to justification of the system and estimate of the impact of the candidate system on the user and organization are documentation and evaluated by management as a step towards implementation. The final report prior to the implementation phases includes procedural flowcharts, record layouts and workable plan for implementing the candidate system.

#### **3.1 DATA FLOW DIAGRAM (DFD)**

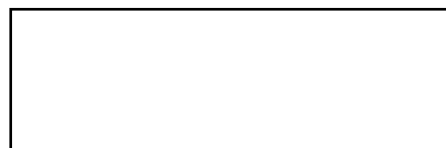
A data flow diagram (DFD) or a bubble chart is a graphical tool for structured analysis. DFD models a system by using external entities from which data flow to a process, which transforms the data and creates output data flows which go other process or external entities or files. Data in files may also flow to processes as inputs.

DFDs can be hierarchically organized, which help in partitioning and analyzing large systems. As a first step, one dataflow diagram can depict an entire system which gives the system overview. It is called context diagram or level 0 DFD. The context diagram can be further expanded.

The successive expansion of a DFD from the context diagram to those giving more details is known as leveling of DFD. Thus a top down approach is used, starting with an overview and then working out the details. The main merit of the DFD is that it can provide an overview of what data a system would process, what transformation of data are done, what files are used, and where the results flow

## DFD DESIGN NOTATION

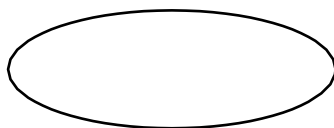
In DFD, there are four main symbols:



External Entity



Data Flow

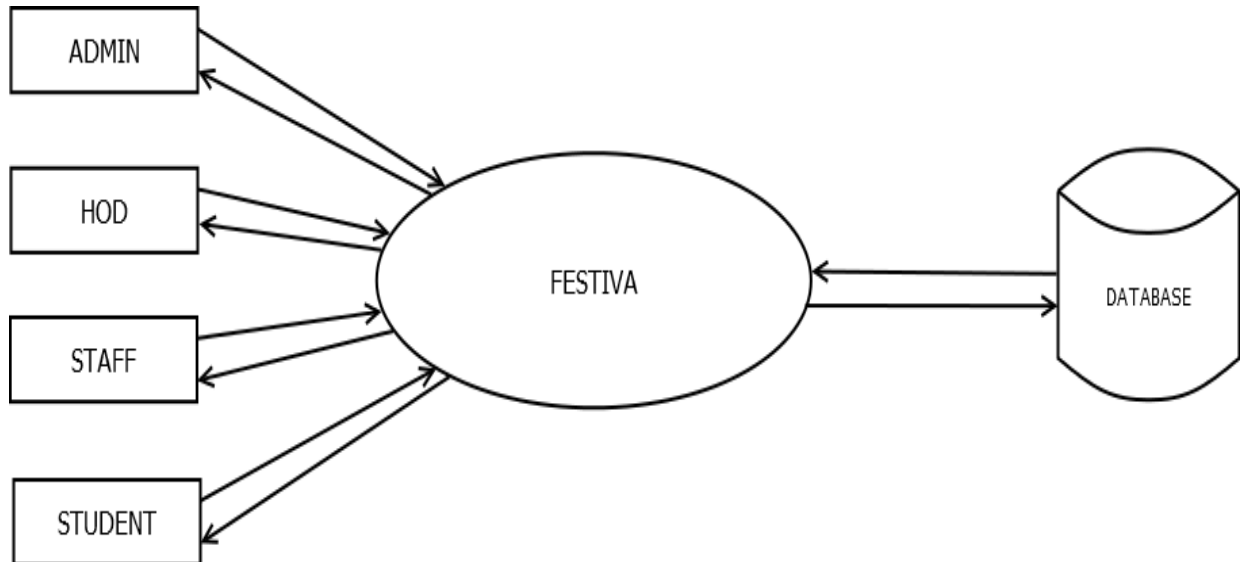


Process

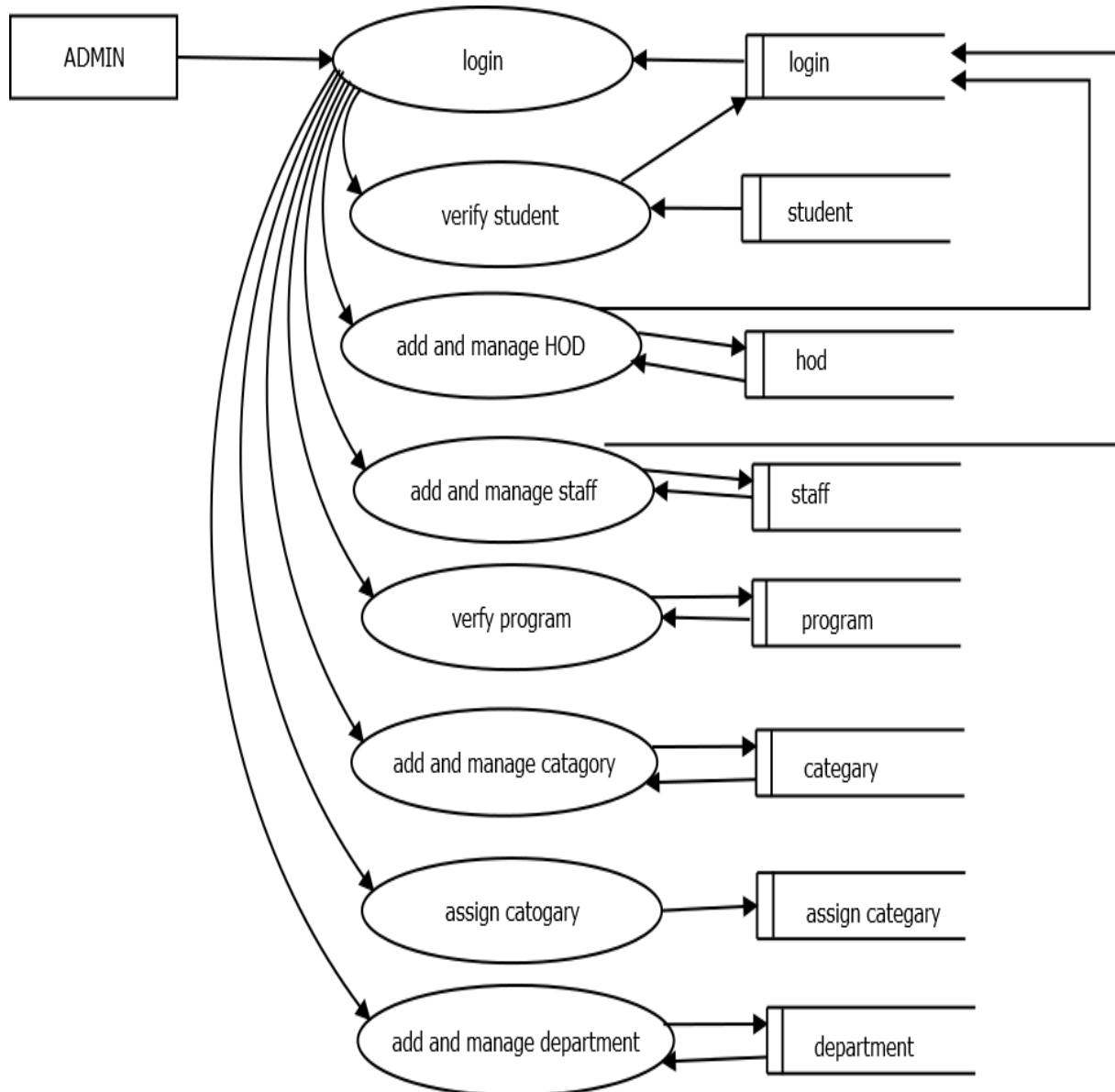


Table

Level 0

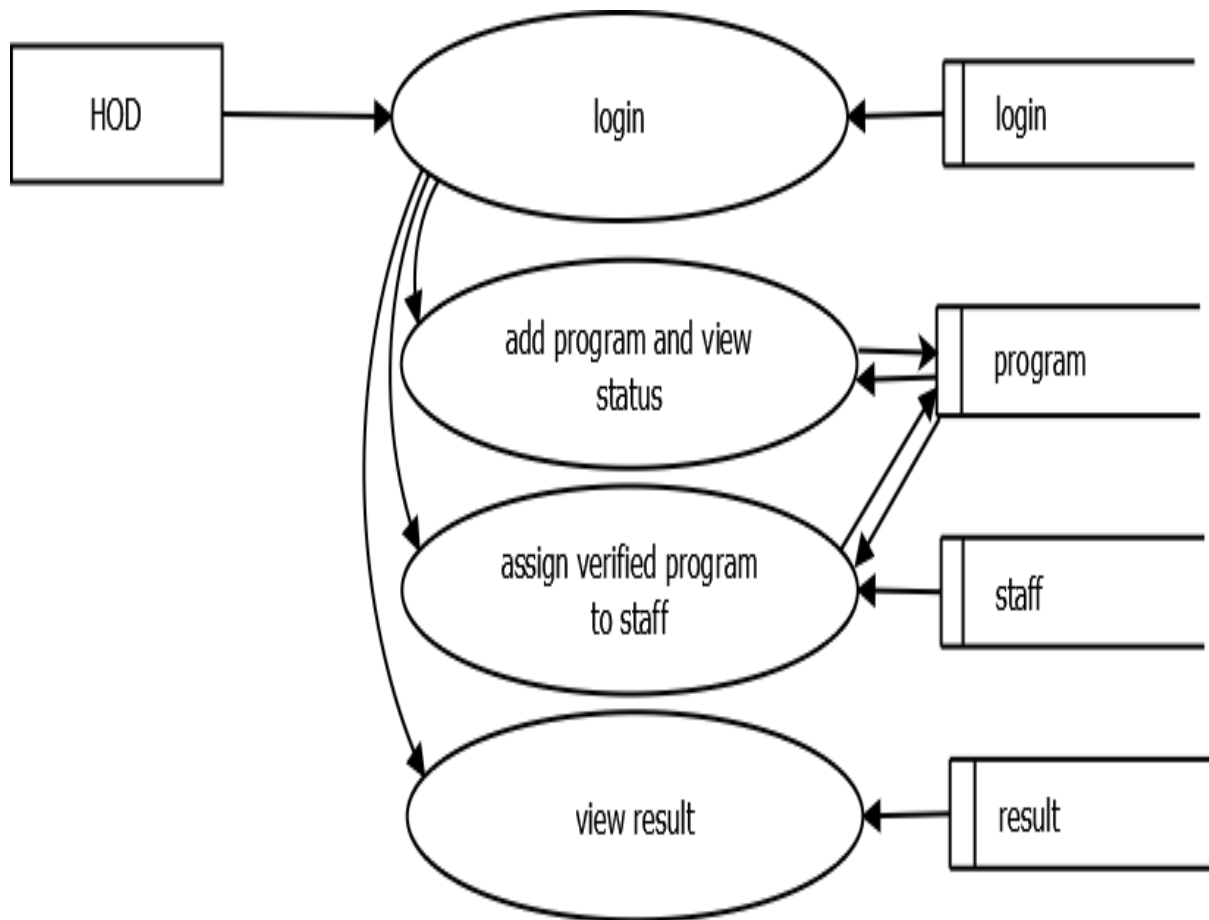


## Level 1.1:Admin

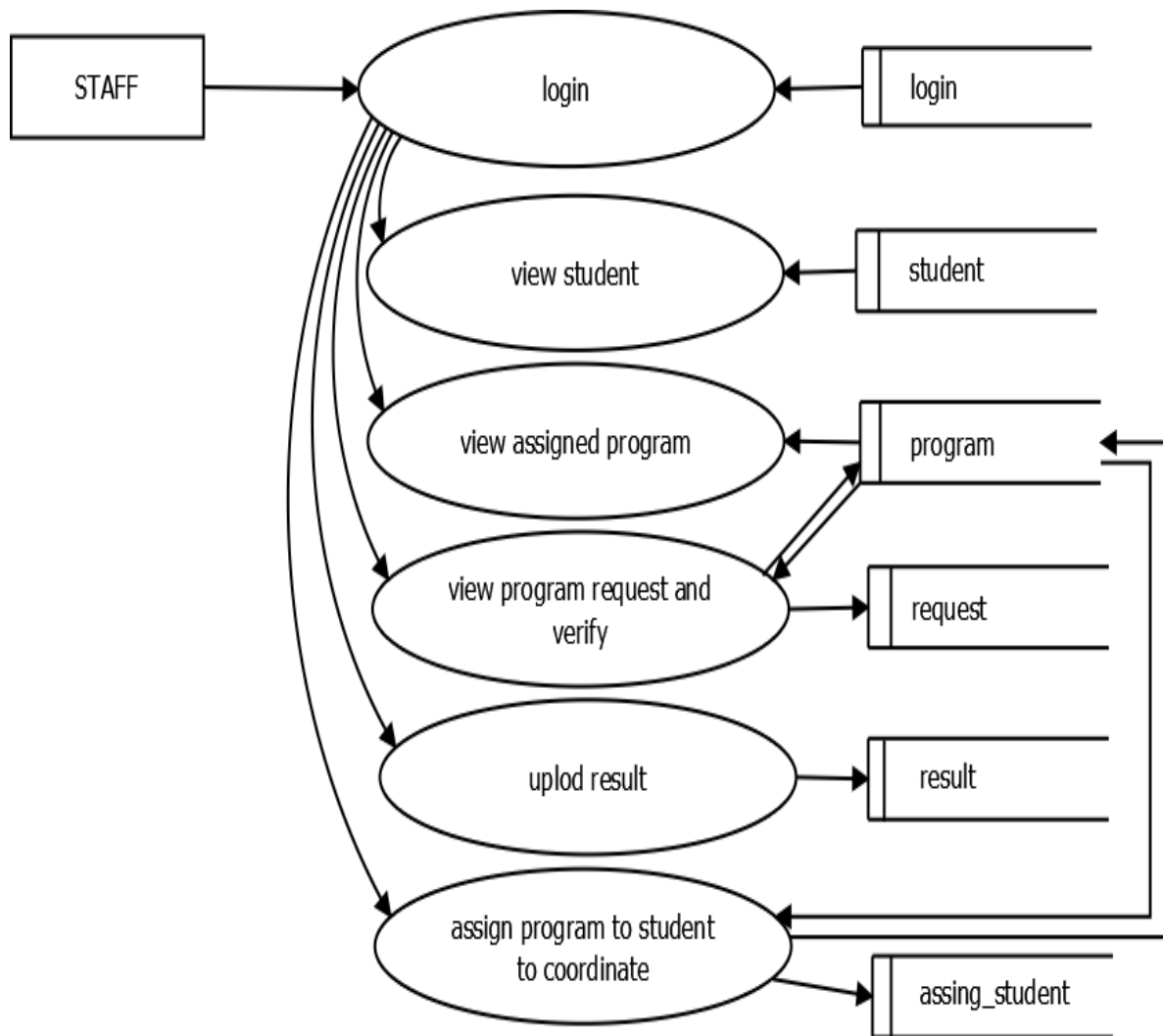




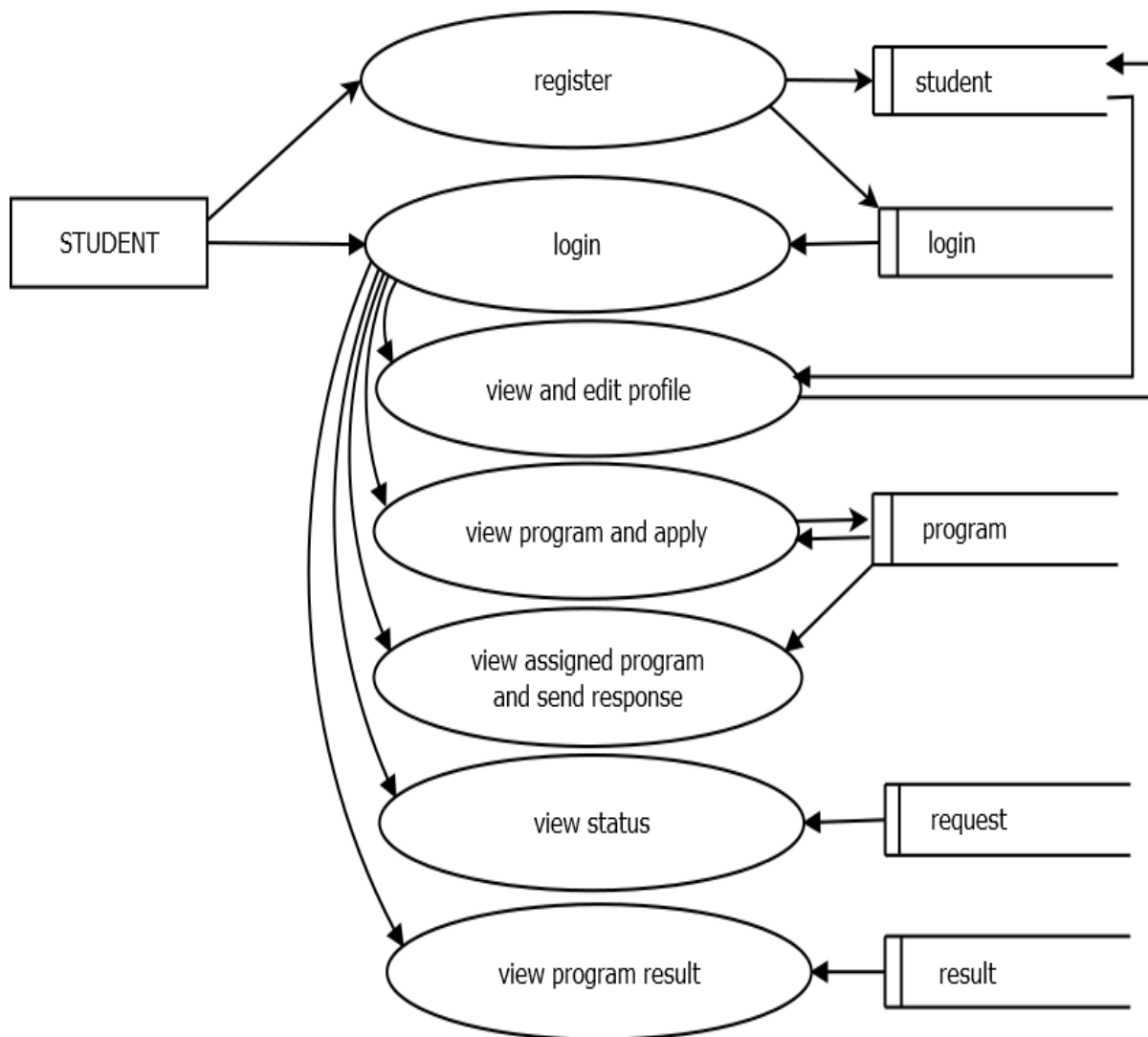
## Level 1.2:HOD



## Level 1.3:Staff



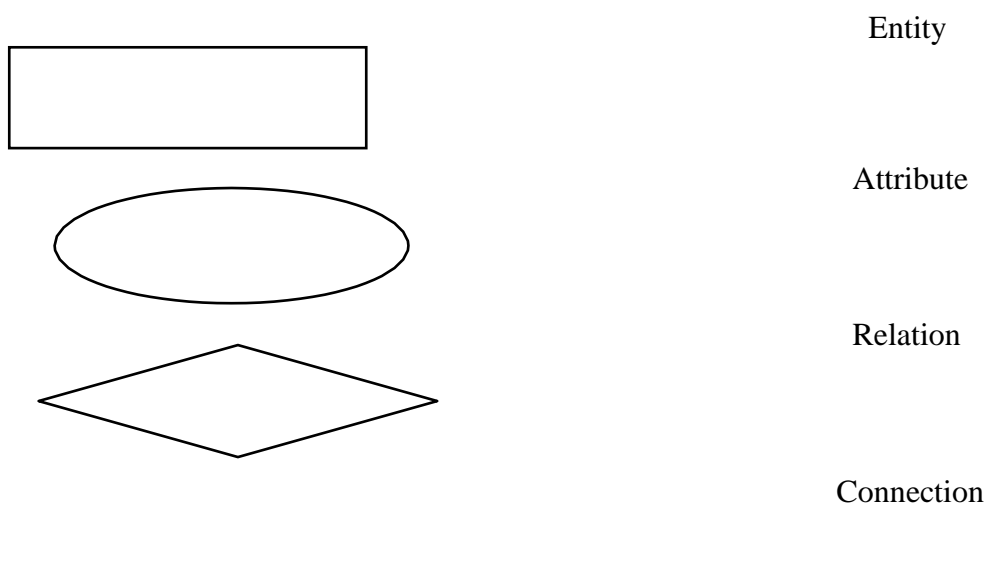
## Level 1.4:Student

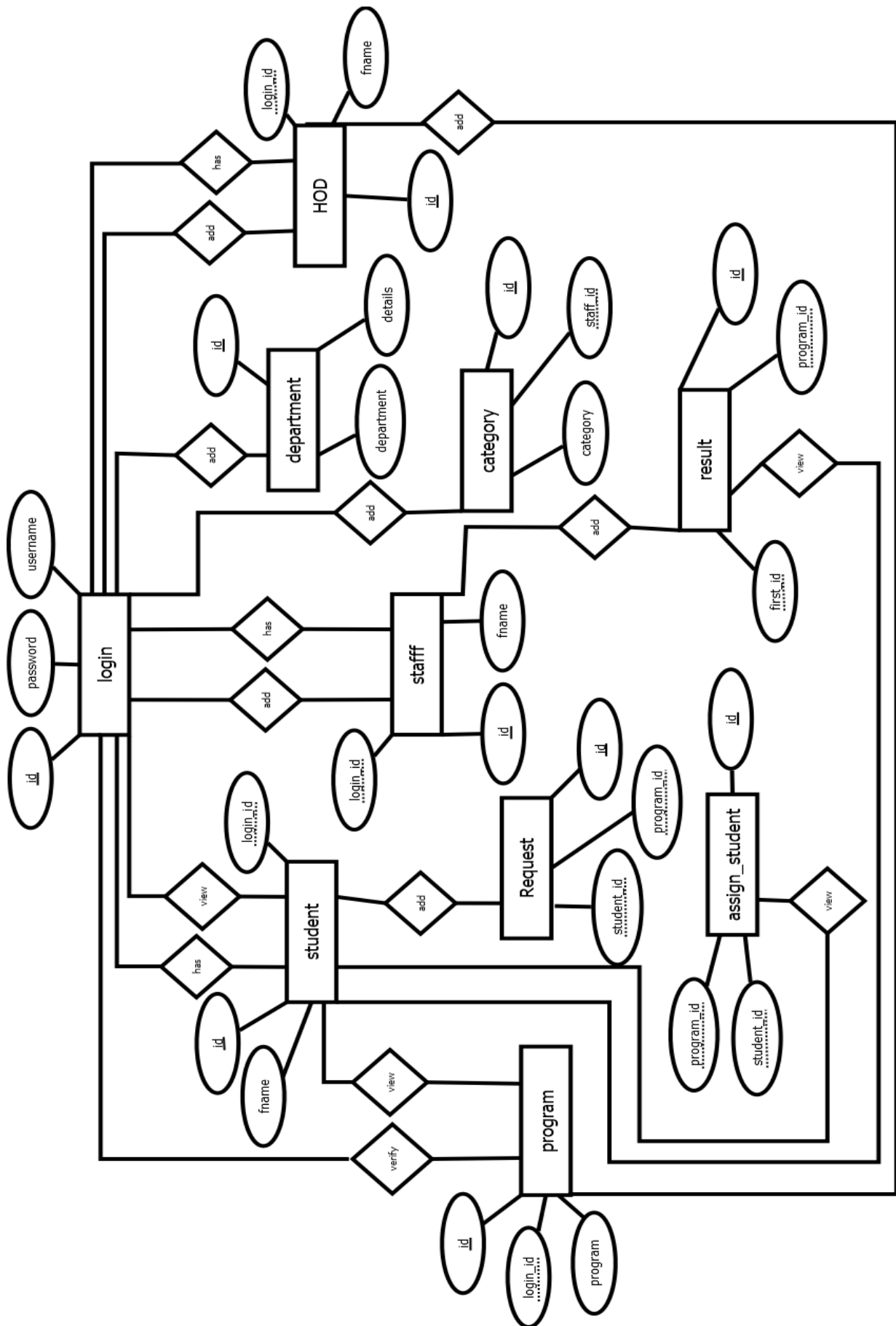


## 3.2 ER DIAGRAM

An ER diagram can express the overall structure of the database graphically. ER diagrams are simple and clear. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

### ENTITY RELATIONSHIP DIAGRAM NOTATIONS





### 3.3 DATABASE DESIGN

A database refers to the collection of data (files and records) which is stored and can be manipulated using software named Database Management Systems. In the previous days where no technical developments took place, data was stored in form of files due to its drawbacks of storage and manipulation, files were replaced by DBMS. Its accuracy of extracting data from queries was better than files. The transaction system worked better in the software. Due to its large storage space, it was chosen over files. Consider an instance, to construct a building, we need a blueprint which makes the work simpler and easier. Likewise, a database needs a blueprint to store the data in a proper sorted way. A database design is a blueprint or a design to store and manipulate the data. The database design is implemented using DBMS. It is important to have a good database design. The reasons are:

- Without a good database design, the database is likely to be unsatisfactory.
- A good database design must be implemented in such ways that the queries are written in a simple and easier manner.
- A good database design doesn't have data redundancies (data redundancy refers to duplication of data.).
- The accuracy must be good enough after the implementation of good database design

#### • Normalization

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divide larger tables into smaller tables and link them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically. The inventor of the relational model Edgar Codd proposed the theory of normalization data with the introduction of the First Normal Form, and he continued to extend theory with Second and Third Normal Form. Later he joined Raymond F. Boyce to develop the theory of Boyce-Codd Normal Form.

### First Normal Form (1NF):

If a relation contains a composite or multi-valued attribute, it violates the first normal form, or the relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is single-valued attribute.

A table is in 1 NF if :

1. There are only Single Valued Attributes.
2. Attribute Domain does not change.
3. There is a unique name for every Attribute/Column.
4. The order in which data is stored does not matter.

### Second Normal Form (2NF):

Second Normal Form (2NF) is based on the concept of full functional dependency. Second Normal Form applies to relations with composite keys, that is, relations with a primary key composed of two or more attributes. A relation with a single-attribute Primary key is automatically in at least 2NF. A relation that is not in 2NF may suffer from the update anomalies.

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, ie, no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

The normalization of 1NF relations to 2NF involves the removal of partial dependencies. If a partial dependency exists, we remove the partially dependent attribute(s) from the relation by placing them in a new relation along with a copy of their determinant .

### Third Normal Form (3NF):

A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form. A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency Third Normal Form (3NF): A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form. A relation is in 3NF if at least one of the following condition holds in every non- trivial function dependency dependencies If a transitive dependency exists, we remove the transitively dependent attribute(s) from the relation by placing the attribute(s) in a new relation along with a copy of the determinant.

### Boyce-Codd Normal Form (BCNF):

Boyce-Codd Normal Form (BCNF) is based on functional dependencies that take into account all candidate keys in a relation; however, BCNF also has additional constraints compared with the general definition of 3NF. A relation is in BCNF if, X is super key for every functional dependency (FD)  $X \rightarrow Y$  in given relation. To determine the highest normal form of a given relation R with functional dependencies, the first step is to check whether the BCNF condition holds. If R is found to be in BCNF, it can be safely deduced that the relation is also in 3NF, 2NF and 1NF as the hierarchy shows. The 1NF has the least restrictive constraint-it only requires a relation R to have atomic values in each tuple. The 2NF has a slightly more restrictive constraint.



### 3.4 TABLE DESIGN

Table 1: login\_table

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/>	id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	username	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	password	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	type	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 2: program\_table

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/>	id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	program	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	details	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	date	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	currentdate	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	status	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	category_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	hod_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	staff_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	poster	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 3: department\_table

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/>	id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	department	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	details	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 5: category\_table

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/>	id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	category	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	staff_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 5: hod\_table

<input type="checkbox"/> Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/> id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> fname	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> lname	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> place	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> post	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> pin	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> gender	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> email	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> phone	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> image	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> department_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> login_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 6: staff\_table

<input type="checkbox"/> Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/> id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> fname	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> lname	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> place	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> post	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> pin	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> gender	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> email	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> phone	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> image	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> department_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> login_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 7: assign\_to\_student\_table

<input type="checkbox"/> Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/> id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> status	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> date	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> program_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> student_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 8: request\_table

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/>	id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	status	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	date	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	program_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	student_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 9: student\_table

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/>	id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	fname	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	lname	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	place	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	post	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	pin	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	gender	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	email	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	phone	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	image	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	department_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	login_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table 10: result\_table

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
<input type="checkbox"/>	id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	first_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	program_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	second_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	third_id	bigint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 3.5 USER INTERFACE DESIGN

The user interface (UI) is the point at which human users interact with a computer, website or application. The goal of effective UI is to make the user's experience easy and intuitive, requiring minimum effort on the user's part to receive maximum desired outcome. UI is created in layers of interaction that appeal to the human senses (sight, touch, auditory and more). They include both input devices like keyboard, mouse, track pad, microphone, touch screen, fingerprint scanner, e-pen and camera and output devices like Monitors, speakers and printers. Devices that interact with multiple senses are called "multimedia user interfaces". For example, everyday UI uses a combination of tactile input (keyboard and mouse) and a visual and auditory output (monitor and speakers).

### 3.6 INPUT DESIGN

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR OMR etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties-

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy It should be easy to fill and straightforward.
- . It should focus on user's consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding-
  - o What are the inputs needed for the system?
  - o How end users respond to different elements of forms and screens.

The objectives of input design are -

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

### Data Input Methods

It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users on the PCs.

A system should prevent user from making mistakes by-

- Clear form design by leaving enough space for writing legibly.
- Clear instructions to fill form.
- Clear form design.
- Reducing key strokes.
- Immediate error feedback.

Some of the popular data input methods are-

- Batch input method (Offline data input method)
- Online data input method Computer
- Computer Readable forms.
- Interactive data input

## 3.7 OUTPUT DESIGN

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

The objectives of Output design are:-

- To develop output design that serves the intended purpose and eliminates the production of unwanted output to develop the output design that meets the end users requirements. .
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person. .
- To make the output available on time for making good decisions.

## **SYSTEM TESTING AND IMPLEMENTATION**

## **4. SYSTEM TESTING AND IMPLIMENTATION**

### **4.1 SYSTEM TESTING**

Testing is an activity to verify that a correct system is being built and is performed with the intent of finding faults in the system. However not restricted to being performed after the development phase is complete, but this is to carry out in parallel with all stages of system development, starting with requirements specification. Testing results, once gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required. A project is said to be incomplete without proper testing.

Testing is an activity to verify that a correct system is being built and is performed with the intent of finding faults in the system. However not restricted to being performed after the development phase is complete, but this is to carry out in parallel with all stages of system development, starting with requirements specification. Testing results, once gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required. A project is said to be incomplete without proper testing.

### **4.2 TESTING METHEDOLOGIES**

- ☐ Unit Testing
- ☐ Integration Testing
- ☐ Validation Testing
- ☐ Output Testing
- ☐ User Acceptance Testing



**Unit Testing**

The first level of testing is called as unit testing. Here the different modules are tested and the specification produced during design for the modules. Unit testing is essential for verification of the goal and to test the internal logic of the modules. Unit testing is conducted to different modules of the project. Errors were noted down and corrected down immediately and the program clarity was increased. The testing was carried out during the programming stage itself. In this step each module is found to be working satisfactory as regard to be expected out from the module

**Integration Testing**

The second level of testing includes integration testing. It is a systematic testing of constructing structure. At the same time tests are conducted to uncover errors with the interface. It need not to be the case, that software whose modules when run individually showing results will also show perfect results when run as a whole. The individual modules are tested again and the results are verified. The goal is to see if the modules integrated between the modules. This testing activity can be considered as testing the design and emphasizes on testing modules interaction.

**Validation Testing**

The next level of testing is validation testing. Here the entire software is tested. The reference document for this process is the requirement and the goal is to see if the software meets its requirements. The requirement document reflects and determines whether the software functions as the user expected. At culmination of integration testing, software is completely assembled as a package and corrected and a final series of software test validation test begins. The proposed system under construction has been tested by using validation testing and found to be working satisfactory. Data validation checking is done to see whether the corresponding entries made in different tables are done correctly. Proper validation checks are done in case of insertion and updating of tables, in order to see that no duplication of data has occurred. If any such case arises proper warning message will be displayed. Double configuration is done before the administrator deletes a data in order to get positive results and to see that o data have been deleted by accident.

**Output Testing**

The output of the software should be acceptable to the system user. The output of requirement is defined during the system analysis. Testing of the software system is done against the output and the output testing was completed with success.

**User Acceptance Testing**

An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system procedures operate to system specification and the integrity of the vital data is maintained.

**4.3 SYSTEM IMPLEMENTATION**

System implementation is the final phase i.e., putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. Implementation involves the conversion of a basic application to complete replacement with a computer system. It is the process of converting to a new or revised system design into an operational one. During the design phase, the products structure, its undergoing data structures, the general algorithms and the interfaces and control/data linkages needed to support communication among the various sub structures were established

Implementation process is simply a translation of the design abstraction into the physical realization, using the language of the target architecture. 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 manual or automated system, or it may be major modification to an existing system. In either case, proper implementation is essential to provide a reliable system to meet organizational requirements. There are three types of implementation:

- ☐ Implementation of a computer system to replace a manual system.
- ☐ Implementation of a new computer system to replace an existing one.
- ☐ Implementation of a modified application to replace an existing one, using the same computer.

The common approaches for implementation are:

### **Parallel Conversion**

In parallel conversion the existing system and new system operate simultaneously until the project team is confident that the new system is working properly. The outputs from the old system continue to be distributed until the new system has proved satisfactorily. parallel conversion is a costly method because of the amount of duplication involved.

### **Direct Conversion**

Under direct conversion method the old system is discontinued altogether and the new system becomes operational immediately. A greater risk is associated with direct conversion is no backup in the in the case of system fails.

### **Pilot Conversion**

A pilot conversion would involve the changing over of the part of the system either in parallel or directly. Use of the variation of the two main methods is possible when part of the system can be treated as a separate entity.

### **User Training**

After the system is implemented successfully, training of the user is one of the most important subtasks of the developer. For this purpose user manuals are prepared and handed over to the user to operate the developed system. Thus the users are trained to operate the developed system. In order to put new application system into use, the following activities were taken care of:

- ☐ Preparation of user and system documentation.
- ☐ Conducting user training with demo and hands on.
- ☐ Test run for some period to ensure smooth switching over the system

The major implementation procedures are:

- ☐ Test plans
- ☐ Training

- ☐ Conversion

### **Test Plans**

The implementation of a computer based system requires that the test data be prepared and the system and its elements be tested in a structured manner.

### **Training**

The purpose of training is to ensure that all the personal who are to be associated with the computer based business system possesses the necessary knowledge skills .As the system provides user friendliness only basic training is needed.

### **Conversion**

It is the process of performing all of the operations that results directly in the turn over of the new system to the user. Conversion has two parts: The creation of a conversion plan at the start of the development phase and the implementation of the plan throughout the development phase. The creation of a system change over plan at the end of the development phase and the implementation of the plan at the beginning of the operation phase.

## **4.4 SYSTEM MAINTANANCE**

The maintenance is an important activity in the life cycle of a software product. Maintenance includes all the activities after the installation of software that is performed to keep the system operational. The maintenance phase of a software life cycle is the time period in which a product performs useful work

Maintenance is classified into four types:

- ☐ Corrective Maintenance
- ☐ Adaptive Maintenance
- ☐ Perfective Maintenance
- ☐ Preventive Maintenance

**Corrective Maintenance**

Corrective maintenance refers to changes made to repair defects in the design, coding, or implementation of the system. Corrective maintenance is often needed for repairing processing or performance failures or making changes because of previously uncorrected problems or false assumptions. Most corrective maintenance problems surface soon after the installation. When corrective maintenance problems surface, they are typically urgent and need to be resolved to curtail possible interruptions in normal business activities.

**Adaptive Maintenance**

Adaptive maintenance involves making changes to an information system to evolve its functionality or to migrate it to different operating environment. Adaptive maintenance is usually less urgent than corrective maintenance because of business and technical changes typically occur some period of time.

**Perfective Maintenance**

Perfective maintenance involves making enhancements to improve processing performance, interface usability, or to add desired, but not necessarily required, system features. Many system professionals feel that perfective maintenance is not really the maintenance but new development.

**Preventive Maintenance**

Preventive maintenance is the only maintenance activity which is carried out without formal maintenance request from the user. When a software company or maintenance agency realizes that the methodologies used in a program have become obsolete, it may decide to develop or modify parts of the program, which do not confirm to the current trends. Of these types, more time and money is spending on perfective than on corrective and adaptive maintenance together.

## **FUTURE ENHANCEMENT**

## **FUTURE ENHANCEMENT**

Festiva, as a program organizing app, stands to enhance organizational efficiency by seamlessly integrating with existing systems such as student information, learning management, and financial management platforms. This integration would streamline data flow, reducing manual entry and optimizing resource allocation. Moreover, Festiva could offer advanced analytics and reporting functionalities, empowering organizers with insights into program performance and attendee engagement. Incorporating gamification elements like points, badges, and leaderboards could incentivize participation and foster a sense of accomplishment among users. Additionally, AI-powered recommendations could assist organizers in curating personalized program suggestions based on user preferences and past attendance. Overall, Festiva has the potential to revolutionize event planning and management by leveraging technology to enhance user experience and streamline organizational processes.

## CONCLUSION



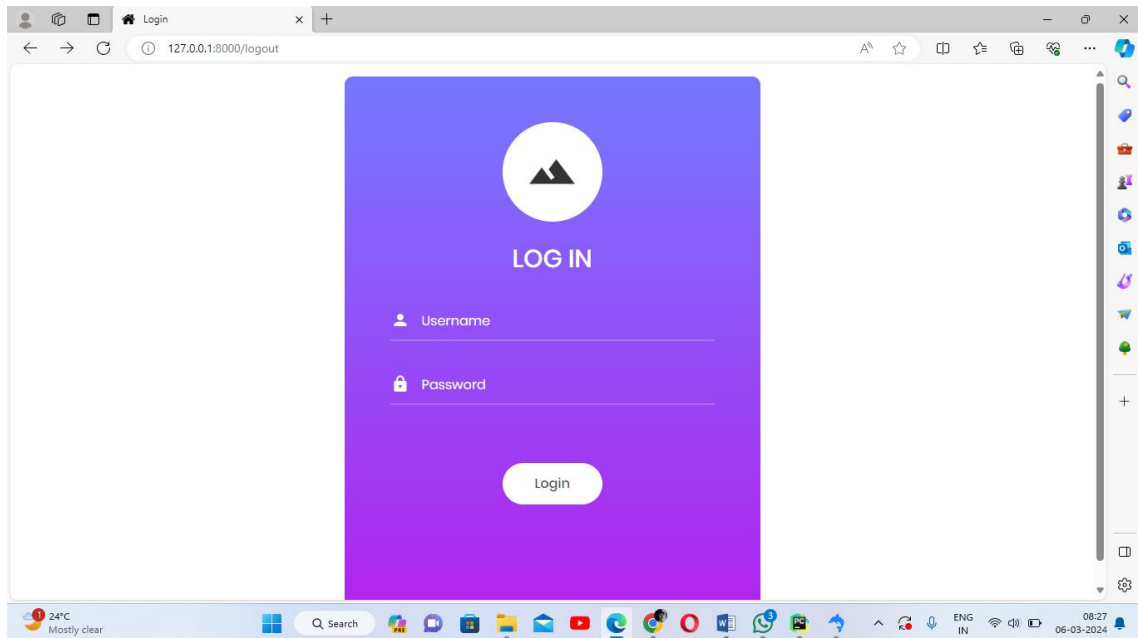
## CONCLUSION

In conclusion, Festiva stands as a cornerstone for academic excellence by transforming program management into a collaborative and efficient endeavor. Its meticulously designed modules empower stakeholders at every level. Administrators can leverage Festiva's robust framework to maintain program details, manage user access, and tailor permissions. Department heads benefit from a dedicated module that simplifies program assignment, facilitates progress monitoring, and empowers faculty evaluation through an intuitive interface. Faculty and staff, meanwhile, gain access to essential tools for managing volunteer activities and inputting results, ensuring the smooth operation of program-related tasks. Finally, students are equipped with a personalized module granting them easy access to program information, a streamlined enrollment process, and the ability to closely track their performance reports. This multifaceted approach fosters transparency, collaboration, and ultimately, a learning environment primed for success. By streamlining program management from the ground up, Festiva empowers colleges and universities to achieve new heights of operational efficiency.

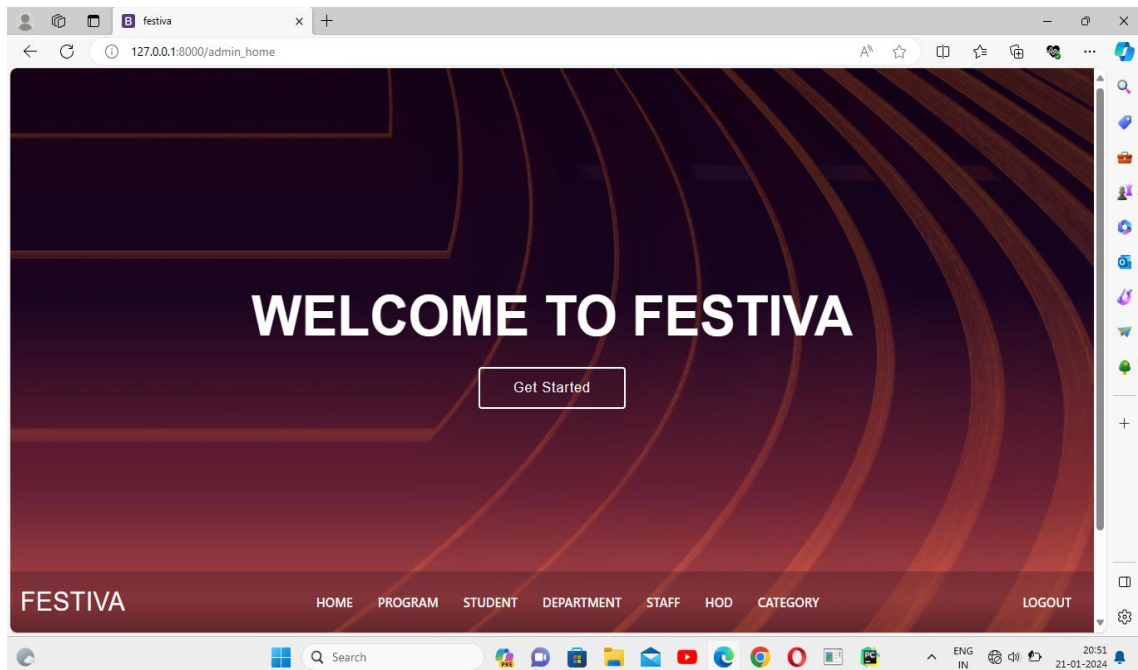
## **APPENDIX I**

## APPENDIX I:SCREENSHOT

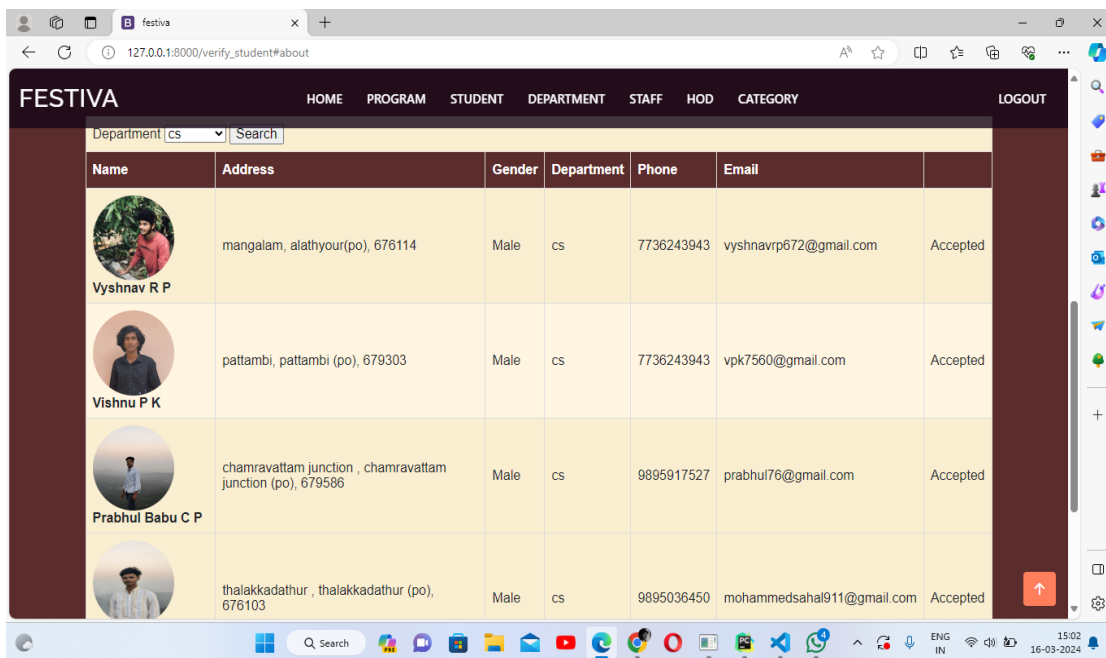
### Login



### Admin-Home







## View And Manage Student

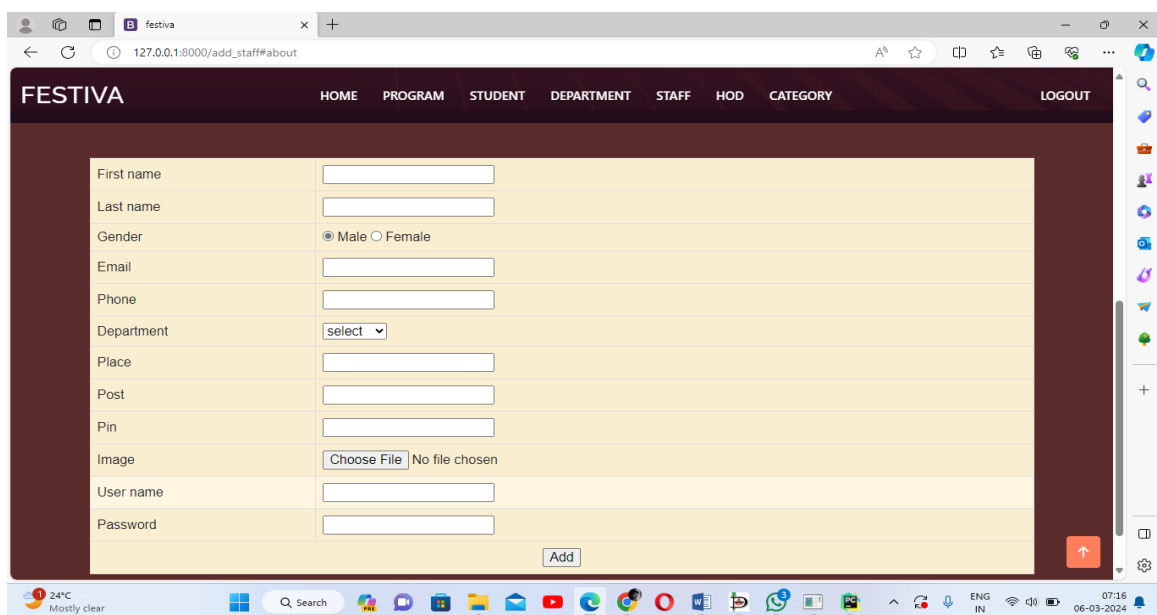


**FESTIVA** | HOME | PROGRAM | STUDENT | DEPARTMENT | STAFF | HOD | CATEGORY | LOGOUT

Department: **cs** | Search

Name	Address	Gender	Department	Phone	Email	
 Vyshnav R P	mangalam, alathyour(po), 676114	Male	cs	7736243943	vyshnavrp672@gmail.com	Accepted
 Vishnu P K	pattambi, pattambi (po), 679303	Male	cs	7736243943	vpk7560@gmail.com	Accepted
 Prabhul Babu C P	chamravattam junction , chamravattam junction (po), 679586	Male	cs	9895917527	prabhul76@gmail.com	Accepted
 thalakkadathur , thalakkadathur (po), 676103		Male	cs	9895036450	mohammedsahal911@gmail.com	Accepted

## Add Saff



**FESTIVA** | HOME | PROGRAM | STUDENT | DEPARTMENT | STAFF | HOD | CATEGORY | LOGOUT

127.0.0.1:8000/add\_staff#about

First name:

Last name:

Gender: ☒ Male ☐ Female

Email:

Phone:

Department:

Place:

Post:

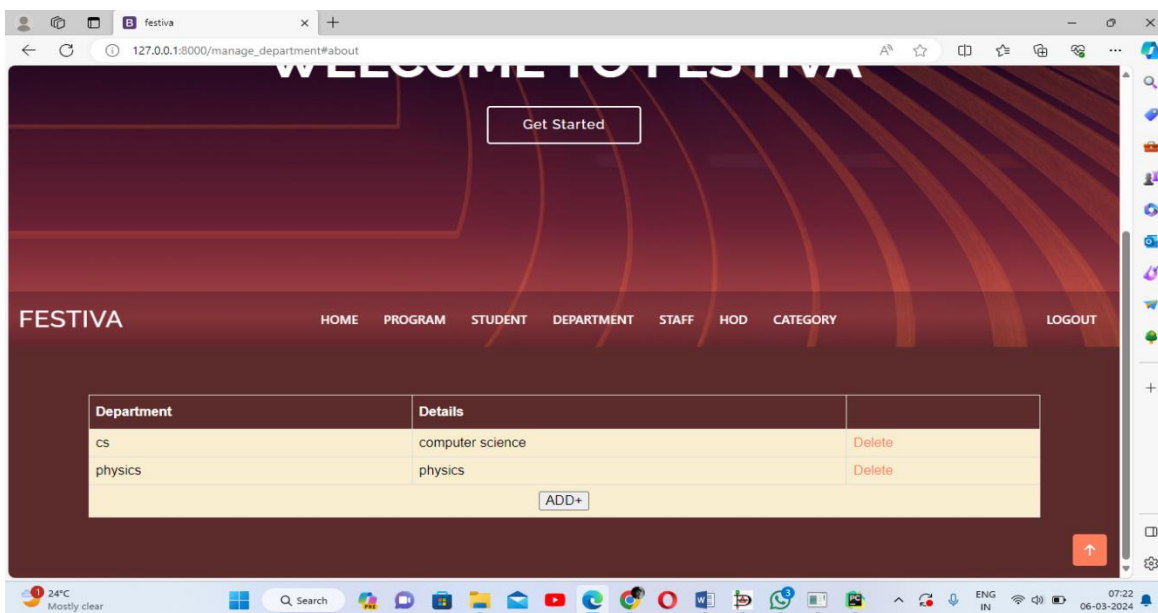
Pin:

Image:  No file chosen

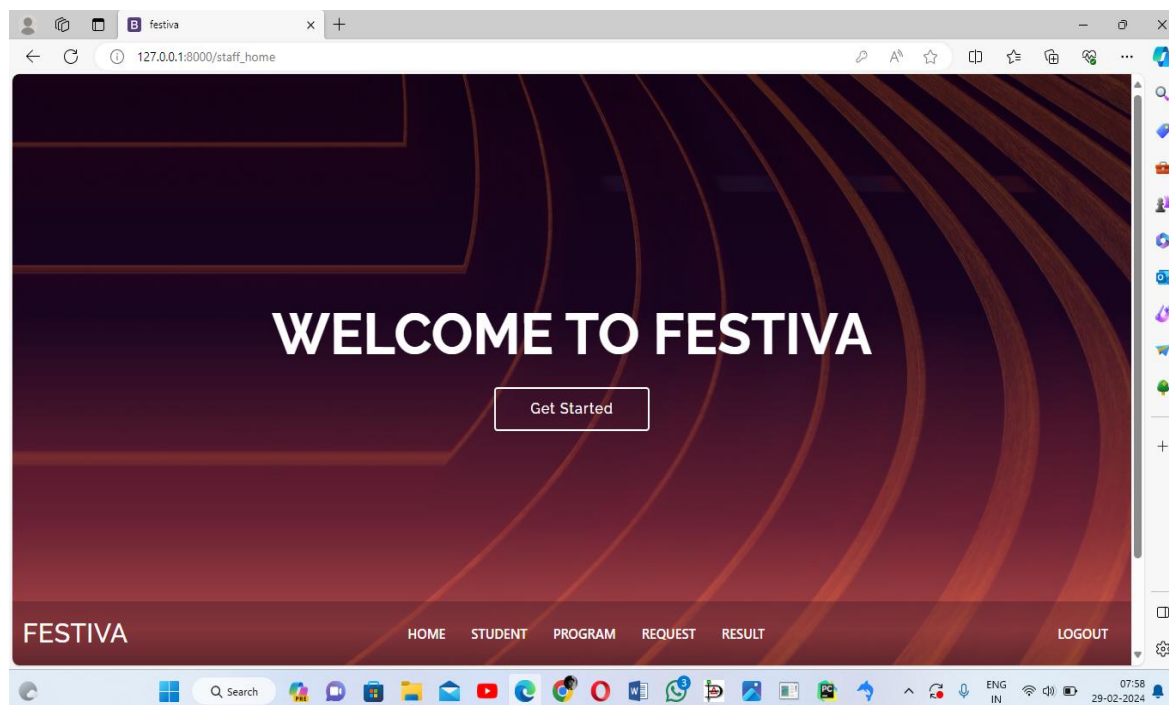
User name:

Password:

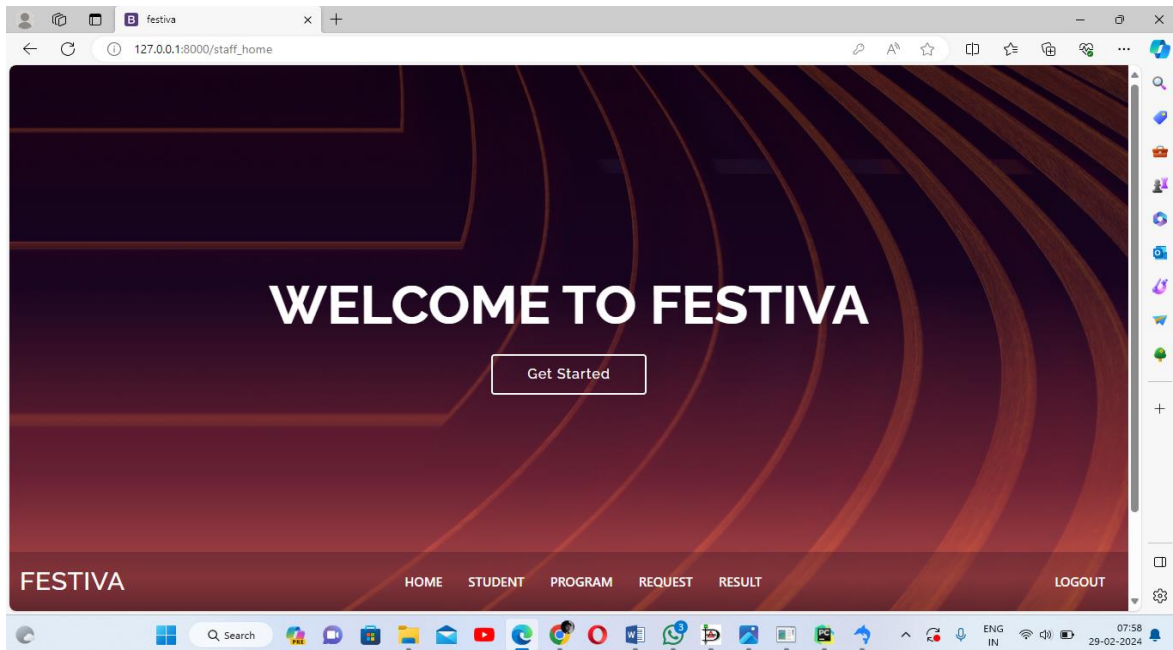
## Manage Department



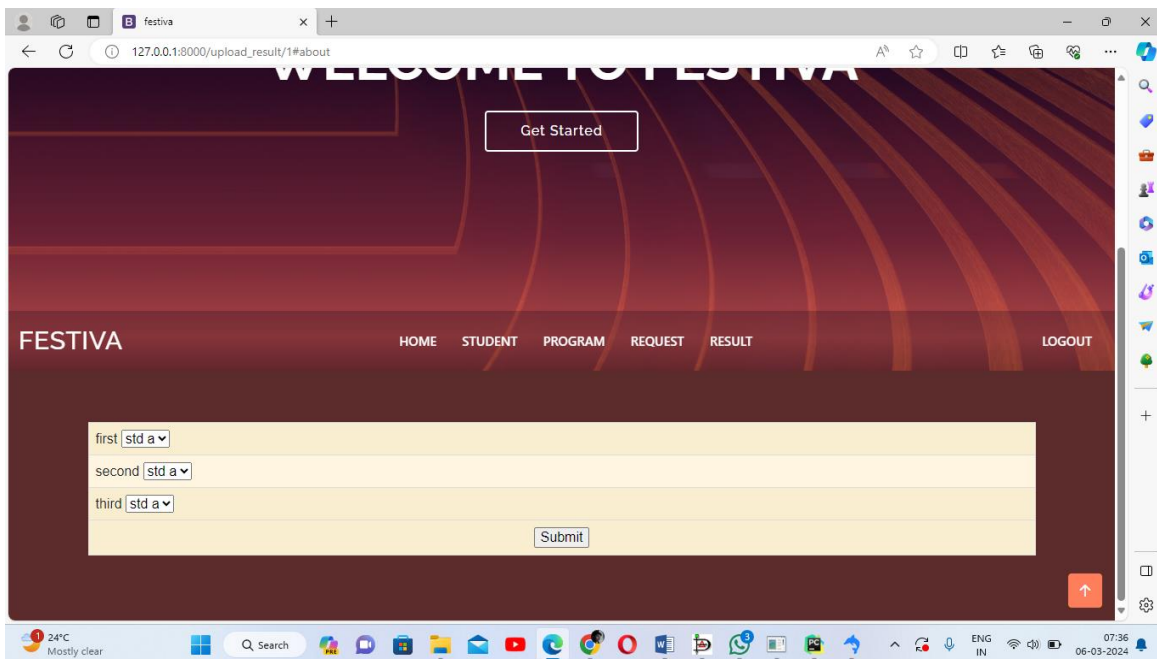
## Staff Home



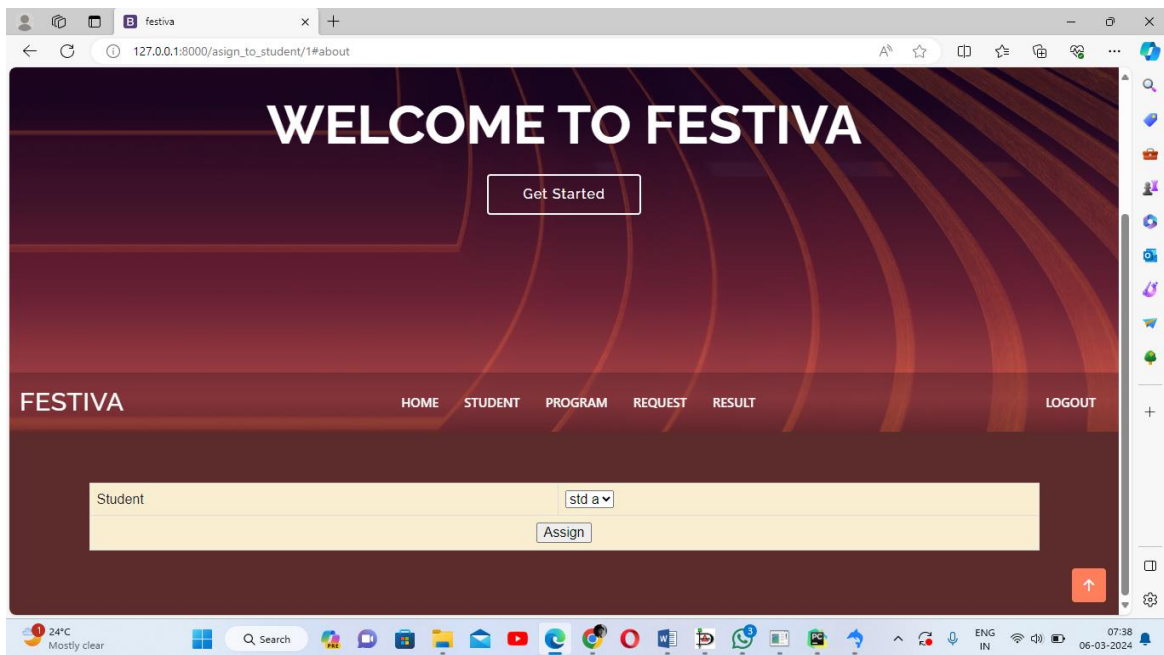
## Manage Request



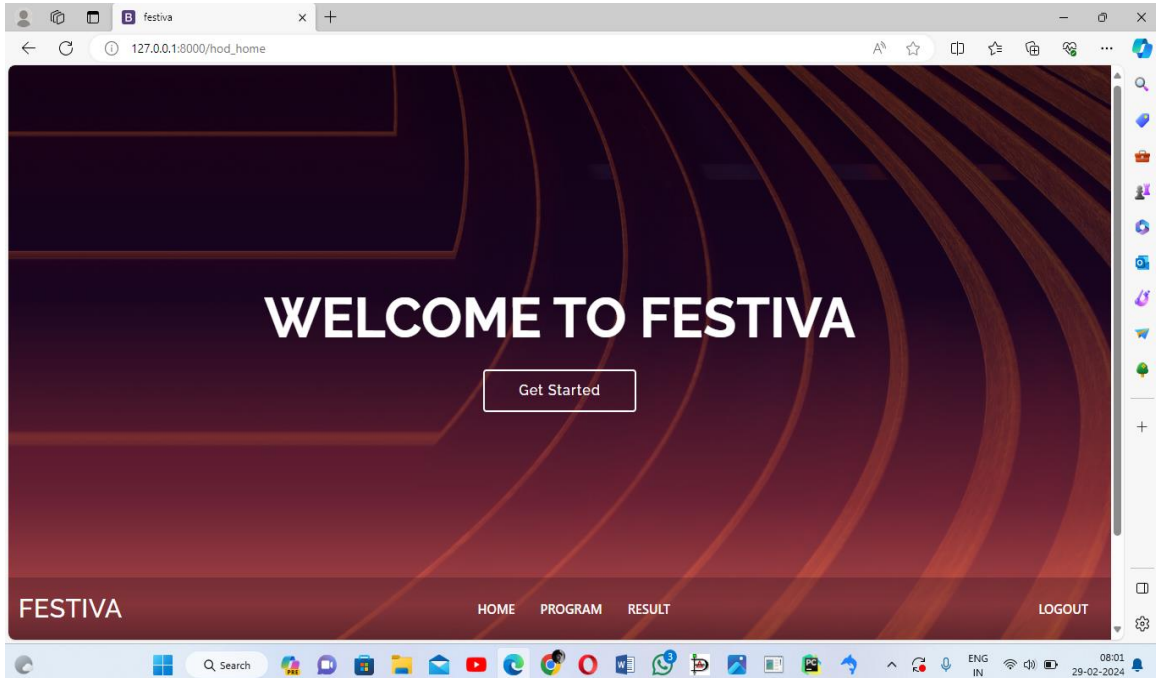
## Upload Result



## Assign to student



## Hod Home





## Manage Program

The screenshot shows the 'FESTIVA' web application with the 'PROGRAM' tab selected. The interface includes a search bar for categories and a table listing programs. The table has columns for Program, HOD, Details, Date, Category, and actions. Two programs are listed: 'olympia' and 'rugby league'.

Program	HOD	Details	Date	Category	
olympia	hod a	college ground	March 16, 2024	sports	Upload result Assign to student
rugby league	hod a	college ground	Feb. 19, 2024	sports	Upload result Assign to student

## Add program

The screenshot shows the 'FESTIVA' web application with the 'PROGRAM' tab selected. The 'Add program' form is displayed, featuring input fields for Program, Details, Date, Category, Staff, and Poster, along with an 'Add' button.



## Edit Program

Program

Details

Date

Category

Staff

Poster  No file chosen

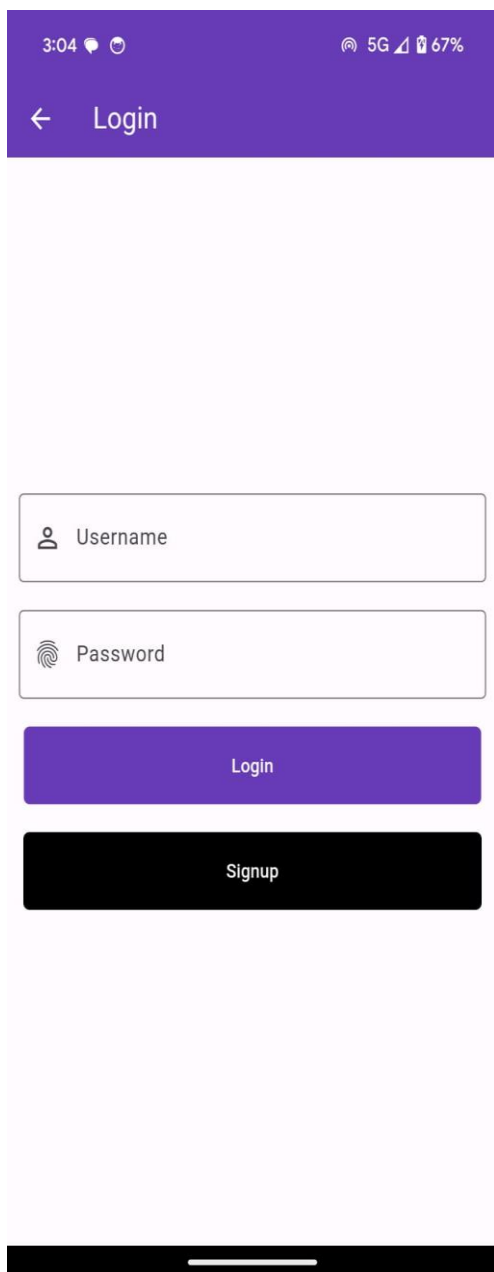
## View Result

Program

program	Details	Result
pgm1	details	1 std a 2 std c 3 std a

## Android

### User login



3:04 5G 67%

← Login

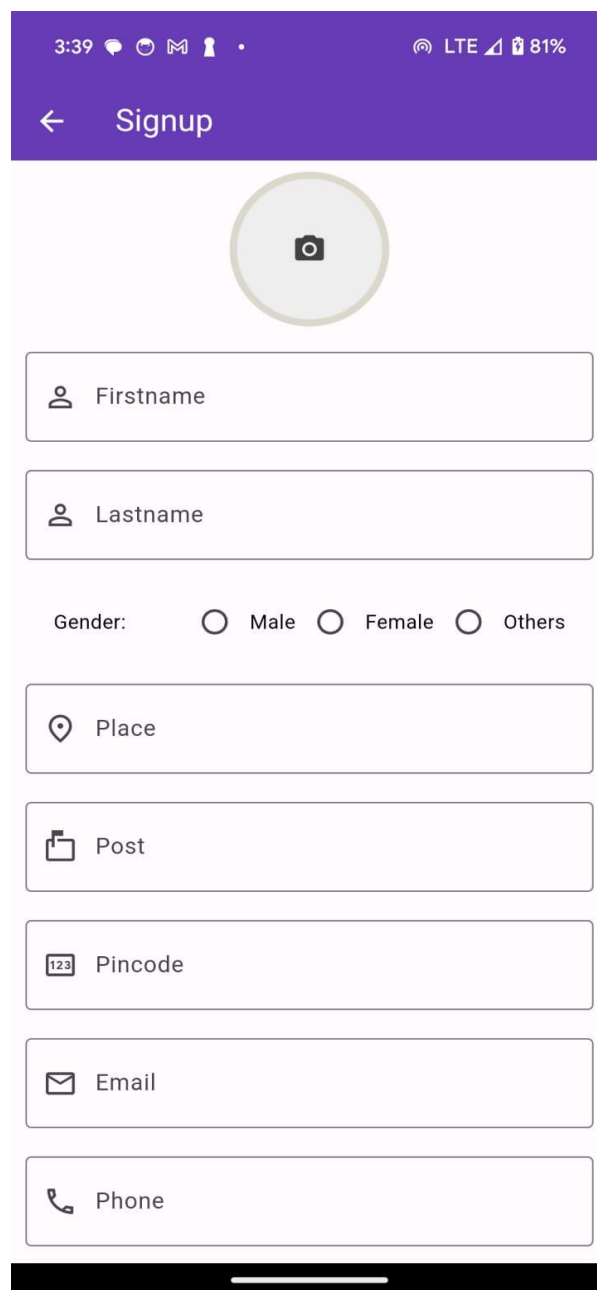
Username

Password

Login

Signup

### Registration



3:39 LTE 81%

← Signup

Profile picture placeholder

Firstname

Lastname

Gender: ☐ Male ☐ Female ☐ Others

Place

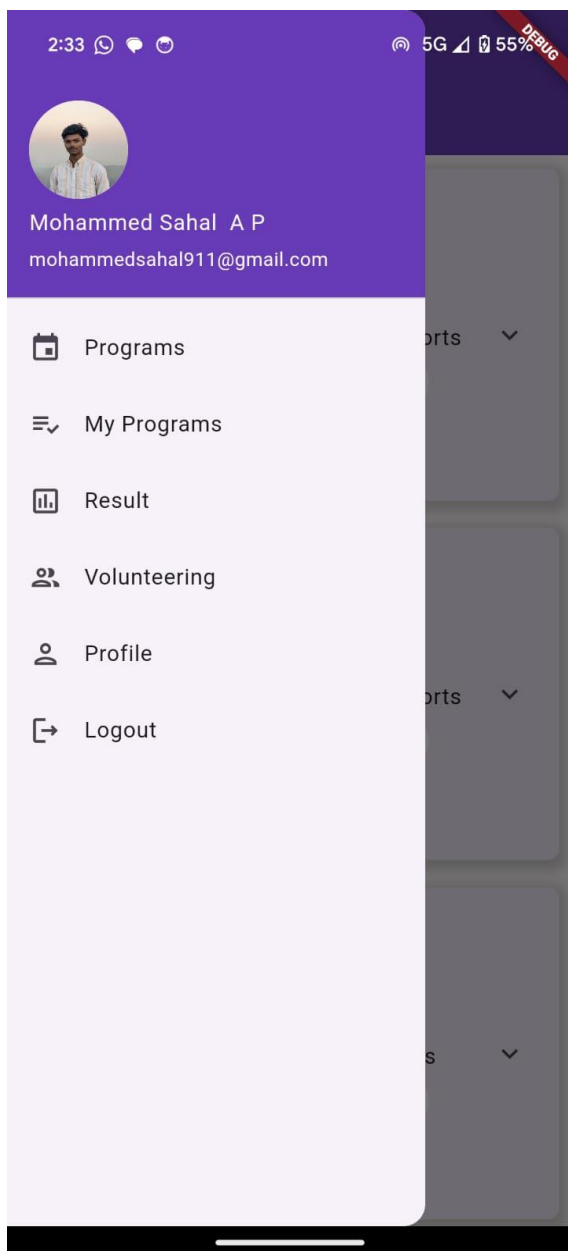
Post

Pincode

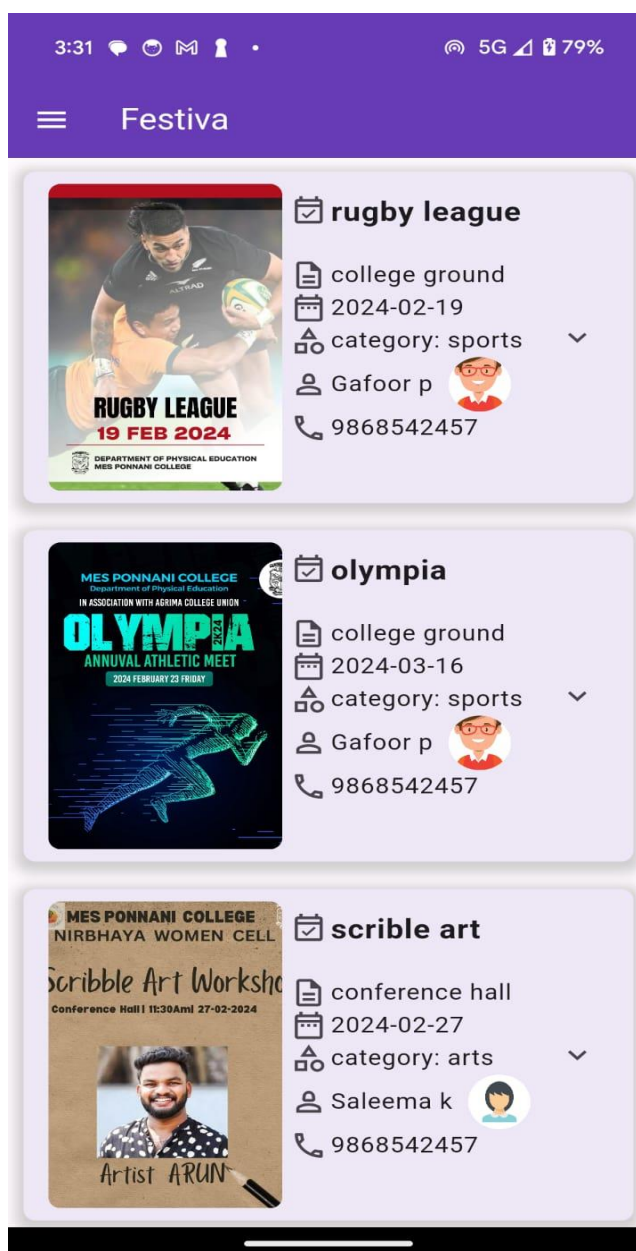
Email

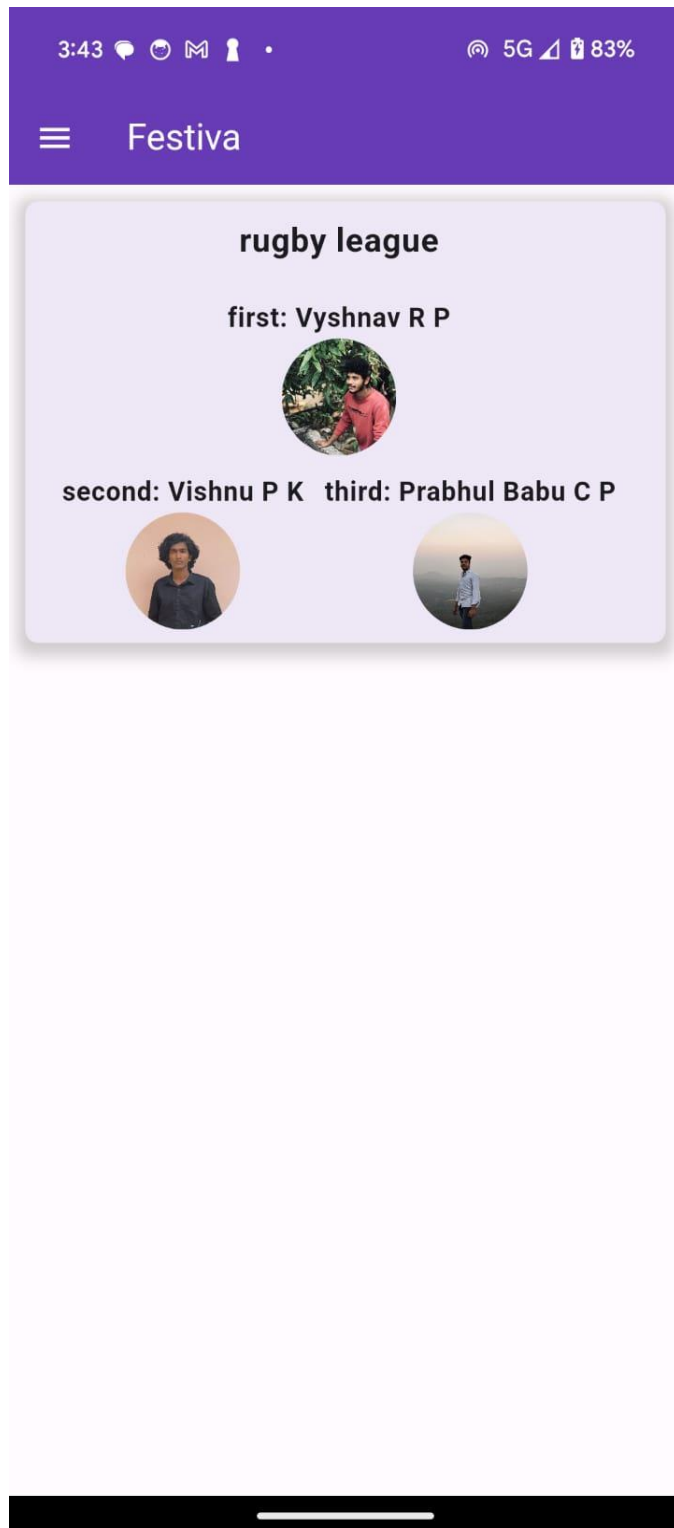
Phone

## Drawer



## programs



**Result**

## **APPENDIX II**

## Appendix II:Source

### codeWeb :login

```
{% load static %}<html lang="en">

<head>

    <title>Login</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

<!--
=====
=====-->

    <link rel="icon" type="image/png" href="{% static 'login/images/icons/favicon.ico' %}"/>

<!--
=====
=====-->

    <link rel="stylesheet" type="text/css" href="{% static
'login/vendor/bootstrap/css/bootstrap.min.css' %} ">

<!--
=====
=====-->

    <link rel="stylesheet" type="text/css" href="{% static 'login/fonts/font-awesome-
4.7.0/css/font-awesome.min.css' %} ">

<!--
=====
=====-->

    <link rel="stylesheet" type="text/css" href="{% static 'login/fonts/ionic/css/material-design-
ionic-font.min.css' %} ">

<!--
=====
=====-->

    <link rel="stylesheet" type="text/css" href="{% static 'login/vendor/animate/animate.css' %} ">

<!--
```

```

=====
=====-->

<link rel="stylesheet" type="text/css" href="{% static 'login/vendor/css-
hamburgers/hamburgers.min.css' %}">

<!--
=====
=====-->

<link rel="stylesheet" type="text/css" href="{% static
'login/vendor/animations/css/animations.min.css' %}">

<!--
=====
=====-->

<link rel="stylesheet" type="text/css" href="{% static 'login/vendor/select2/select2.min.css'
%}">

<!--
=====
=====-->

<link rel="stylesheet" type="text/css" href="{% static
'login/vendor/daterangepicker/daterangepicker.css' %}">

<!--
=====
=====-->

<link rel="stylesheet" type="text/css" href="{% static 'login/css/util.css' %}">

<link rel="stylesheet" type="text/css" href="{% static 'login/css/main.css' %}">

<!--
=====
=====-->

</head>

<body>

<div class="limiter">

<div class="container-login100" style="background-image: url('images/bg-01.jpg');">

<div class="wrap-login100">

<form class="login100-form validate-form" action="login_post" method="POST">

<span class="login100-form-logo">

<i class="zmdi zmdi-landscape"></i>

```

</span>

<span class="login100-form-title p-b-34 p-t-27">

Log in

</span>

<div class="wrap-input100 validate-input" data-validate = "Enter username">

<input class="input100" type="text" name="textfield" placeholder="Username">

<span class="focus-input100" data-placeholder="&#xf207;"></span>

</div>

<div class="wrap-input100 validate-input" data-validate="Enter password">

<input class="input100" type="password" name="textfield2"  
placeholder="Password">

<span class="focus-input100" data-placeholder="&#xf191;"></span>

</div>

<div class="container-login100-form-btn">

<button class="login100-form-btn">

Login

</button>

</div>

<div class="text-center p-t-90">

<a class="txt1" href="#">

</a>

</div>



```

        </form>

    </div>

</div>

</div>

<div id="dropDownSelect1"></div>

<!--
=====
=====-->

    <script src="{% static 'login/vendor/jquery/jquery-3.2.1.min.js' %}"></script>

<!--
=====
=====-->

    <script src="{% static 'login/vendor/animstition/js/animstition.min.js' %}"></script>

<!--
=====
=====-->

    <script src="{% static 'login/vendor/bootstrap/js/popper.js' %}"></script>
    <script src="{% static 'login/vendor/bootstrap/js/bootstrap.min.js' %}"></script>

<!--
=====
=====-->

    <script src="{% static 'login/vendor/select2/select2.min.js' %}"></script>

<!--
=====
=====-->

    <script src="{% static 'login/vendor/daterangepicker/moment.min.js' %}"></script>
    <script src="{% static 'login/vendor/daterangepicker/daterangepicker.js' %}"></script>

<!--
=====
=====-->

    <script src="{% static 'login/vendor/countdowntime/countdowntime.js' %}"></script>

<!--
=====
=====

```

```
=====-->
```

```
<script src="{% static 'login/js/main.js' %}"></script>
```

```
</body>
```

```
</html>
```

## Add Staff

```
{% extends 'admin/admin_index.html' %}
```

```
{% block body %}
```

```
<form action="add_staff_post" method="post" enctype="multipart/form-data" name="form1" id="form1">
```

```
<table width="200" border="1" id="customers">
```

```
<tr>
```

```
<td>First name </td>
```

```
<td><input type="text" name="textfield" required pattern="[A-Z a-z . ]{2,}" /></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Last name </td>
```

```
<td><input type="text" name="textfield2" required pattern="[A-Z a-z . ]{1,}" /></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Gender</td>
```

```
<td><input name="radiobutton" type="radio" value="Male" checked />
```

```
Male
```

```
<input name="radiobutton" type="radio" value="Female" />
```

```
Female</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Email</td>
```

```
<td><input type="email" name="textfield3" required/></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Phone</td>
```

```
<td><input type="text" name="textfield4" required pattern="[6789][0-9]{9}" /></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Department</td>
```

```
<td><select name="select" required>
```

```
<option value="">select</option>
```

```
{% for i in val %}
```

```
<option value="{{ i.id }}">{{ i.department }}</option>
```

```
{% endfor %}
```

```
</select></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Place</td>
```

```
<td><input type="text" name="textfield5" required pattern="[A-Z a-z . ]{2,}" /></td>
```

```
</tr>
```

```

<tr>
  <td>Post</td>
  <td><input type="text" name="textfield6" required pattern="[A-Z a-z . ]{2,}" /></td>
</tr>
<tr>
  <td>Pin</td>
  <td><input type="text" name="textfield7" required pattern="[0-9]{6}" /></td>
</tr>
<tr>
  <td>Image</td>
  <td><input type="file" name="file" required /></td>
</tr>
<tr>
  <td>User name </td>
  <td><input type="text" name="textfield8" required/></td>
</tr>
<tr>
  <td>Password</td>
  <td><input type="text" name="textfield9" required/></td>
</tr>
<tr>
  <td colspan="2"><div align="center">
    <input name="Submit" type="submit" value="Add" />
  </div></td>
</tr>
</table>
</form>
{% endblock %

```

## Android

### Main

```
import 'package:flutter/material.dart';
import 'package:sample_project/ipset.dart';

void main() {
  runApp(const festiva());
}
class festiva extends StatelessWidget {
  const festiva({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        elevatedButtonTheme: ElevatedButtonThemeData(style: ElevatedButton.styleFrom())
        //
      ),
      home: ipset(),
    );
  }
}
```

### Signup

```
import 'dart:convert';
import 'dart:io';

import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:sample_project/login.dart';

import 'package:shared_preferences/shared_preferences.dart';
import 'package:http/http.dart' as http;

class registration extends StatefulWidget {
  const registration({super.key});
```

```

@override
State<registration> createState() => _registrationState();
}
// enum Gender { male, female, other }
// enum Department { it, hr, finance, marketing, operations }

class _registrationState extends State<registration> {

  final TextEditingController firstnameController = TextEditingController();
  final TextEditingController lastnameController = TextEditingController();
  final TextEditingController placeController = TextEditingController();
  final TextEditingController postController = TextEditingController();
  final TextEditingController pinController = TextEditingController();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController phoneController = TextEditingController();
  final TextEditingController usernameController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  String? selectedGender;
  List _deparment=[];
  String? _currentdepartment;
  final _formKey = GlobalKey<FormState>();
  XFile? _image;

  _imgFromCamera() async {
    XFile? image = await ImagePicker().pickImage(
      source: ImageSource.camera,
      imageQuality: 50,
    );
    if (image != null) {
      setState() {
        _image = image;
      });
    }
  }

  _imgFromGallery() async {
    XFile? image = await ImagePicker().pickImage(
      source: ImageSource.gallery,
    );
    if (image != null) {
      setState() {
        _image = image;
      });
    }
  }

  void _showPicker(context) {
    showModalBottomSheet(
      context: context,

```

```

builder: (BuildContext bc) {
  return SafeArea(
    child: Container(
      child: new Wrap(
        children: <Widget>[
          new ListTile(
            leading: new Icon(Icons.photo_library),
            title: new Text('Photo Library'),
            onTap: () {
              _imgFromGallery();
              Navigator.of(context).pop();
            },
          ),
          new ListTile(
            leading: new Icon(Icons.photo_camera),
            title: new Text('Camera'),
            onTap: () {
              _imgFromCamera();
              Navigator.of(context).pop();
            },
          ),
        ],
      ),
    ),
  );
}

getdepartment() async{
  try {
    final sh = await SharedPreferences.getInstance();
    String url = sh.getString("url").toString();
    final response = await http.get(Uri.parse(url + "get_department"));
    if (response.statusCode == 200)
    {
      var data= json.decode(response.body);
      setState(() {
        _deparment = data['data'];
      });
    }
    else {
      throw Exception('Failed to fetch department');
    }
    // print(datalist);
    // print(_deparment);
  }
  catch (error) {
    // Handle errors here
    print(error);
  }
}

```

```
@override
void initState() {
  super.initState();
  getdepartment();
}
```

```
@override
Widget build(BuildContext context) {
```

```
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.deepPurple,
      title: const Text("Signup",
        style: TextStyle(
          color: Colors.white, // Text color
        ),
      ),
      iconTheme: IconThemeData(
        color: Colors.white, // Icon color
      ),
      elevation: 4.0, // Elevation
    ),
    body: SafeArea(
      child: Center(
        child: Form(
          key: _formKey,
          child: SingleChildScrollView(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                Padding(
                  padding: EdgeInsets.all(8.0),
                  child: Center(
                    child: GestureDetector(
                      onTap: () {
                        _showPicker(context);
                      },
                      child: CircleAvatar(
                        radius: 55,
                        backgroundColor: Color(0xffdbd8cd),
                        child: _image != null
                          ? ClipRRect(
                              borderRadius: BorderRadius.circular(50),
                              child: Image.file(
                                File(_image!.path),
                                width: 100,
                                height: 100,
                                fit: BoxFit.cover,
```





```

    },
  ),
),
Padding(
  padding: const EdgeInsets.all(10.0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      const Text(
        "Gender:      ",
        textAlign: TextAlign.left,
        style: TextStyle(color: Colors.black),
      ),
      Radio<String>(
        value: 'Male',
        groupValue: selectedGender,
        onChanged: (value) {
          setState() {
            selectedGender = value;
          });
      },
    ),
    Text(
      'Male',
      style: TextStyle(color: Colors.black),
    ),
    Radio<String>(
      value: 'Female',
      groupValue: selectedGender,
      onChanged: (value) {
        setState() {
          selectedGender = value;
        });
      },
    ),
    Text(
      'Female',
      style: TextStyle(color: Colors.black),
    ),
    Radio<String>(
      value: 'Others',
      groupValue: selectedGender,
      onChanged: (value) {
        setState() {
          selectedGender = value;
        });
      },
    ),
    Text(
      'Others',
      style: TextStyle(color: Colors.black),

```

```

    ),
  ],
),
),

//    Padding(
// padding: const EdgeInsets.all(8.0),
// child: Column(
//   crossAxisAlignment: CrossAxisAlignment.start,
//   children: [
//     Text(
//       'Select your department:',
//       style: TextStyle(fontSize: 16),
//     ),
//     DropdownButton<Department>(
//       value: selectedDepartment,
//       onChanged: (Department newValue) {
//         setState() {
//           selectedDepartment = newValue;
//         };
//       },
//       items:
Department.values.map<DropdownMenuItem<Department>>((Department value) {
//       return DropdownMenuItem<Department>(
//         value: value,
//         child: Text(value.toString().split('.').last),
//       );
//     }).toList(),
//   ),
//   SizedBox(height: 16),
//   Text(
//     'Selected Department: ${selectedDepartment ?? 'None'}',
//     style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
//   ),
// ],
// ),
// ),

Padding(
  padding: const EdgeInsets.all(10.0),
  child: TextFormField(
    controller: placeController,
    decoration: const InputDecoration(

      border: OutlineInputBorder(),
      prefixIcon: Icon(Icons.place_outlined),
      labelText: "Place",
      hintText: "Place",
    ),
    validator: (value) {
      if (value!.isEmpty) {

```

```

        return 'Please enter your place';
    }
    return null; // Return null if the input is valid
  },
),
),
Padding(
  padding: const EdgeInsets.all(10.0),
  child: TextFormField(
    controller: postController,
    decoration: const InputDecoration(

      border: OutlineInputBorder(),
      prefixIcon: Icon(Icons.markunread_mailbox_outlined),
      labelText: "Post",
      hintText: "Post",
    ),
    validator: (value) {
      if (value!.isEmpty) {
        return 'Please enter your post';
      }
      return null; // Return null if the input is valid
    },
  ),
),
Padding(
  padding: const EdgeInsets.all(10.0),

  child: TextFormField(
    controller: pinController,
    decoration: const InputDecoration(
      prefixIcon: Icon(Icons.pin_outlined),
      border: OutlineInputBorder(),
      labelText: "Pincode",
      hintText: "Pincode",
    ),
    validator: (value) {
      if (value!.isEmpty) {
        return 'Please enter your pincode';
      }
      return null; // Return null if the input is valid
    },
  ),
),
Padding(
  padding: const EdgeInsets.all(10.0),
  child: TextFormField(
    controller: emailController,
    decoration: const InputDecoration(

      border: OutlineInputBorder(),

```

```

        prefixIcon: Icon(Icons.mail_outline_outlined),
        labelText: "Email",
        hintText: "Email",
      ),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Please enter your email';
        }
        return null; // Return null if the input is valid
      },
    ),
  ),
  Padding(
    padding: const EdgeInsets.all(10.0),
    child: TextFormField(
      controller: phoneController,
      decoration: const InputDecoration(

        border: OutlineInputBorder(),
        prefixIcon: Icon(Icons.phone_outlined),
        labelText: "Phone",
        hintText: "Phone",
      ),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Please enter your phone';
        }
        return null; // Return null if the input is valid
      },
    ),
  ),
  Padding(
    padding: const EdgeInsets.only(left: 20.0,
      right: 10.0,
      top: 10.0,
      bottom: 10.0,),
    child: Row(
      children: [
        const Text('Department'),
        const Padding(padding: EdgeInsets.only(left: 30.0,)),
        DropdownButton<String>(
          key: UniqueKey(),
          hint: Text("select"),
          value: _currentdepartment,
          items: _deparment?.map((item){
            return DropdownMenuItem(
              value: item["id"].toString(),
              child: Text(item["name"]),
            );
          }).toList(),

```

```

        onChanged:(String? newValue){

            setState() {
                _currentdepartment=newValue;

            });

        },

    ),

    ],),
),

// child: Column(
//   crossAxisAlignment: CrossAxisAlignment.center,
//   mainAxisAlignment: MainAxisAlignment.center,
//   children: [
//     DropdownButton<String>(
//       key: UniqueKey(),
//       value: selectedName,
//       hint: Text("Select Name"),
//       onChanged: (String? newValue) {
//         setState() {
//           print(newValue);
//           int j=names.indexOf(newValue!);
//           selectedid=login_id[j];
//           print(selectedid);
//           selectedName = newValue;
//         });
//       },
//       items: names.map((String name) {
//         return DropdownMenuItem<String>(
//           value: name,
//           child: Text(name),
//         );
//       }).toList(),
//     ),

//   ],
// ),
// ),

```

```

Padding(
  padding: const EdgeInsets.all(10.0),
  child: TextFormField(
    controller: usernameController,
    decoration: const InputDecoration(
      prefixIcon: Icon(Icons.person_outlined),

```

```

        border: OutlineInputBorder(),
        labelText: "Username",
        hintText: "Username",
      ),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Please enter your username';
        }
        return null; // Return null if the input is valid
      },
    ),
  ),
  Padding(
    padding: const EdgeInsets.all(10.0),
    child: TextFormField(
      controller: passwordController,
      decoration: const InputDecoration(

        border: OutlineInputBorder(),
        prefixIcon: Icon(Icons.fingerprint),
        labelText: "Password",
        hintText: "Password",
      ),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Please enter your password';
        }
        return null; // Return null if the input is valid
      },
    ),
  ),
  Padding(
    padding: EdgeInsets.all(10.0),
    child: TextButton(
      style: ButtonStyle(
        foregroundColor: MaterialStateProperty.all(Colors.white),
        backgroundColor: MaterialStateProperty.all(Colors.deepPurple),
        padding: MaterialStateProperty.all(EdgeInsets.symmetric(horizontal: 170, vertical: 18.0)),
        shape: MaterialStateProperty.all(RoundedRectangleBorder(borderRadius:
BorderRadius.circular(5.0))),
      ),
      onPressed: ()async {
        if(!_formKey.currentState!.validate())
        {print("vvvvvvvvvvvvvvvvvvvvvvvv");}
        else{
          final sh = await SharedPreferences.getInstance();
          String firstname=firstnameController.text.toString();
          String lastname=lastnameController.text.toString();
          String place=placeController.text.toString();
          String post=postController.text.toString();
          String pin=pinController.text.toString();

```

```

String email=emailController.text.toString();
String phone=phoneController.text.toString();
String username=usernameController.text.toString();
String password=passwordController.text.toString();
final bytes = File(_image!.path).readAsBytesSync();
String base64Image = base64Encode(bytes);
String url = sh.getString("url").toString();
print("okkkkkkkkkkkkkkkkkkk");
var data = await http.post(
  Uri.parse(url+"andregister"),
  body: {'firstname':firstname,
        "lastname":lastname,
        "place":place,
        "post":post,
        "pin":pin,
        "gender":selectedGender.toString(),
        "email":email,
        "phone":phone,
        "username":username,
        "password":password,
        "photo":base64Image,
        "department":_currentdepartment.toString(),

        });
var jasondata = json.decode(data.body);
String status=jasondata['task'].toString();
if(status=="ok")
{

  // ignore: use_build_context_synchronously
  Navigator.push(context,
    MaterialPageRoute(builder: (context) => login()));

}
else{
  print("error");

}
},
child: Text("signup"),

),
),
Padding(
padding: const EdgeInsets.all(10.0),
child: TextButton(
style: ButtonStyle(
foregroundColor: MaterialStateProperty.all(Colors.white),
backgroundColor: MaterialStateProperty.all(Colors.black),
padding: MaterialStateProperty.all(EdgeInsets.symmetric(horizontal: 174, vertical: 18.0)),

```

```
    shape: MaterialStateProperty.all(RoundedRectangleBorder(borderRadius:
BorderRadius.circular(5.0))),
  ),
    onPressed: ()async {
Navigator.push(context,
    MaterialPageRoute(builder: (context) => login()));
  },
  child: const Text("Login"),
),
)

],
),
),
),
),
),
),
);
}
}
```