COLLEGE CONNECT (MOBILE APPLICATION FOR COLLEGE COMMUNITY)

PROJECT REPORT

BACHELOR OF SCIENCE COMPUTER SCIENCE

(University of Calicut)

Submitted by

ANAGHARAJ NR

FARSHANA NAZNI A

SHAHABAS AV

SHAHAL MUBASHIR P

SINUSANTH V

ASPIRE COLLEGE OF ADVANCED STUDIES

2021-2024 Academic Year

DEPARTMENT OF COMPUTER SCIENCE

ASPIRE COLLEGE OF ADVANCED STUDIES

THRITHALA-679534

DEPARTMENT OF COMPUTER SCIENCE

ASPIRE COLLEGE OF ADVANCED STUDIES THRITHALA-679534

ASPIRE COLLEGE OF ADVANCED STUDIES,THRITHALA

(AFFILIATED TO UNIVERSITY OF CALICUT)

CERTIFICATE

This is certify that the project report entitled "COLLEGE CONNECT (MOBILE APPLICATION FOR COLLEGE COMMUNITY)" is a bonafide record done by FARASHANA NAZNI A(AGAVBCA002),SHAHAL MUBASHEER P(AGAVBCA014),ANAGHARAJ N.R (AGAVBCA025),SHAHABAS A.V(AGAVBCA037),SINUSANTH V(AGAVBCA038) during the academic year 2021-2024 towards the partial fulfilment of the requirements for the award of BACHELOR OF SCIENCE COMPUTER APPLICATION of university of Calicut.

……. **……**

Project Guide
Mrs. SRUTHY M

Head of Department
Mrs. SRUTHY M

…….. **……..**

Principal
Mr. RIYAS EP

External Examiner

ACKNOWLEDGEMENT

At the outset, we thank the God Almighty for the grace, strength and hope to make our endeavor a success.

We take this opportunity to express our sincere thanks and wholehearted gratitude to our beloved principal **RIYAS E.P**, with great appreciation and endorsement.

We also express our gratitude to **SRUTHY M**, Head of the Department for providing us with adequate facilities, ways and means by which we are able to complete this project work. We express our sincere gratitude to her for his constant support and valuable suggestions

Words are insufficient to express our sincere thanks to **SRUTHY M,** Asst. Prof. Department of Computer science for coordinating and helping us with all the amenities and for his insightful comments and constructive criticism for successful completion of this project.

We express our immense pleasure and thankfulness to all the **Teachers of the Department of Computer Science, Aspire College of Advanced Studies, Thrithala** for their cooperation and support.

We extend our sincere gratitude to **RISS TECHNOLOGIES** to give us an opportunity to carry out a full semester live project.

FARSHANA NAZNI A

SHAHAL MUBASHIR P

ANAGHARAJ N R

SHAHABAS AV

SINUSANTH V

CONTENTS

INTRODUCTION

1.1 ORGANIZATION PROFILE

RISS TECHNOLOGIES Ammankovil Cross Rd , YMCA junction,Ernakulam, Kerala was incorporated in 2013with an objective of offering advanced technological solutions in the fields of information technology (iD and information technology enabled services (ites) including software development, web designing/development, domain registration.web hosting, server administration(windows/linux), technical support, mobile application development, server colocation, search engine optimization, medical transcription and software training. we are now the front-line service provider in it infrastructure services industry, catering to both domestic and international clients, we are proud of our solid team of information systems professionals, including those trained abroad, who study, design, develop, enhance, customize, implement, maintain and support various aspects of it products and services,

Our training team is a team of highly talented professionals. they make technical experts rather than professionals. we can give our students to best start for an it carrier through our embedded or software workshops. we offer good placement assistance to our students. our hard working team coupled with professionalism results in total efficiency and superior skills, which benefit our clients and trainees alike. always we are holding the principle that "teach technology for to use". we are not training students only for getting some certifications or to get some jobs. we want they have to use it and invent something for the technology. that is the reason behind why our students decorate the positions of lot of it companies.

we provide industry-standard, ieee/non ieee on-job live software projects to mtech, btech, be, mse, bse, mca, bea and polytechnic students, using languages and tools like android, python, java and jee, j2me, and php. for these live projects, we maintain strategic alliances with international it majors.

1.2 PROJECT PROFILE

CollegeConnect is an innovative mobile application that aims to create a thriving and interconnected college community by fostering seamless communication and collaboration among students, faculty, and staff. This app is designed to enhance the overall college experience, providing a platform for students to engage, share, and access essential information effortlessly.

Key Features:

- **Campus Networking**: CollegeConnect offers a user-friendly interface that allows students to connect with their peers, faculty members, and campus organizations. It facilitates the formation of study groups, clubs, and interestbased communities, promoting a sense of belonging and camaraderie.

- **Event Management**: The app enables easy organization and promotion of campus events, workshops, seminars, and social gatherings. Students can browse through the event calendar, RSVP, and receive reminders to ensure active participation in various activities.

- **Academic Resources**: CollegeConnect provides a central repository of academic resources, including lecture notes, study materials, and past exam papers. Students can collaborate on projects, seek help, and share

- **Real-Time Notifications**: Stay updated with the latest college announcements, class schedules, and important deadlines through real-time push notifications. This feature ensures that students never miss critical information and can manage their academic responsibilities effectively.

- **Virtual Classrooms**: CollegeConnect integrates virtual classrooms, facilitating remote learning and video conferencing for students and professors. This feature accommodates students who cannot physically attend classes, ensuring an inclusive educational experience.

- **Job and Internship Portal**: The app offers a dedicated job and internship portal where students can explore career opportunities, internships, and campus placements. It allows

- **Campus News and Updates:** CollegeConnect aggregates campus news and updates, providing students with a comprehensive overview of college happenings, achievements, and notable alumni success stories.

- **Campus Safety and Emergency Assistance:** The app includes a safety feature that allows students to alert campus security or emergency services in case of any distress or critical situations, ensuring a safer environment for everyone.

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

1. **Facebook Groups**: Many colleges and universities have official or student-created Facebook groups where students, faculty, and staff can join, share information, and engage in discussions related to campus activities, events, and academics.

2. **WhatsApp Groups**: WhatsApp is widely used for creating groups within college communities. These groups serve as a platform for instant messaging, sharing updates, and coordinating various activities among students and faculty.

3. **Discord:** Discord is commonly used by gaming communities, but it has gained popularity in college settings as well. Students and organizations create servers to chat, voice call, and share information on various topics and events.

4. **GroupMe:** GroupMe is a group messaging app that allows college students to create chat groups for different clubs, classes, or social circles. It enables easy communication and coordination among members.

5. Slack: Although commonly used in workplaces, some college organizations and student groups have adopted Slack to manage communication and collaborate on projects, events, or initiatives.

6. Campus-Specific Apps: Some colleges and universities have developed their own official mobile applications to enhance campus life. These apps often include features such as event calendars, academic resources, campus maps, and notifications for announcements.

7. LinkedIn Campus: LinkedIn offers a feature called "LinkedIn Campus" that allows students to connect with peers, alumni, and faculty within their college or university community. It facilitates professional networking and career development.

2.2 System Study

In this phase, the development team visits the user and studies their system.

They investigate the need for development in the given system. By the end of the Feasibility study, the team furnishes a document that holds the different specific recommendations for the

candidate system. It also includes the personnel assignments, costs, project schedule and target dates.

The requirements-gathering process is intensified and focuses specifically on software. To understand the nature of the program(s) to be built, the System Engineer (developer) must understand the information area for the software, as well as required function, its behaviour, performance and other systems it interfaces with. The crucial purpose of this phase is to find the need and to define the problem that has to be solved.

2.3 Feasibility study

All projects are feasible if they have unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is necessary and prudent to evaluate the feasibility of a project at the earliest possible time Facts considered in the feasibility analysis were.

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility
- Behavioural Feasibility

2.3.1 Technical feasibility

It deals with hardware as well as software requirements. An idea from the outline design to system requirement in terms of inputs, outputs, file and procedure is drawn and the type of hardware and software required for running the system are analyzed.

Keeping in mind of the above considerations, the resource availability at this company was observed. It was found that the company has sufficient resources to develop the current project; hence the system is technically feasible.

2.3.2 Behavioural feasibility

People are inherently resistant to changes and computer is known for facilitating changes. An estimate should be made of how strongly the user staff's react towards the developments of the computerized system.

In the proposed system the manpower is reduced so unnecessary burden is reduced.

Therefore the system is behaviourally feasible.

SYSTEM REGUIREMENTS

3.1 SOFTWARE REQUIREMENTS

A software requirement specification (SRS), a requirements specification for a software system, is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional require-ments. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints) the software requirements specification document enlists all necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with the project team and customer.

Operating System: WINDOWS 8 or above for better performance

Front end: Python (For web application), Android (Mobile Application)

Back end: MYSQL

Software:SubLimeText, WAMP, Android Studio

Web Browser: Internet Explorer/Google Chrome/Firefox

Web Server: Apache

3.2 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

Processor: Intel Pentium or above.

Hard Disc: 320GB.

Display Type: PC Display.

Keyboard: PC/AT Enhanced PS/Keyboard (110/10Key).

Mouse: First/Pilot Mouse Serial (C48).

Input Device: Mouse, keyboard , **Output Device**: Monitor, Mobile Display

SYSTEM DESIGN

SYSTEM DESIGN AND DEVELOPMENT PROCESS

Design is the first step in development phase for every engineering product of system. Computer software designing techniques like engineering design approach in other disciplines, changes, continuously as, new method, better analysis and broader understanding evolves,

System design involves translating information requirement and conceptual design into technical specification and general flow of processing. After the user requirement are identified, related information is gathering to verify the problem, and after evaluating the existing system a new system is proposed. The proposed system consists of various tables, their maintenance and report generation.

System Design

System design involves translating information requirements and conceptual design into technical specification and general flow of processing. After the user requirements are identified, related information is gathered to verify the problem and after evaluating the existing system, a new system is proposed. The proposed system consists of various tables, their maintenance and report generation.

System design is the process of developing specification for a candidate system that meet the criteria established in the system analysis. Major step in system design is the preparation of the input forms and the output reports in a form applicable to the

The main objective of the system design is to use the package easily by any operator. System Design is the creative act of invention, developing new inputs, a database, offline files, method, procedures and output for processing business to meet an organization objective. System design builds information gathered during the system analysis.

input specifications describe the manner in which data enter the system for processing. the design of the input should have

- effectiveness
- accuracy

- simplicity
- ease to use
- consistency
- attractiveness
- Easy Data Input

Data entry has been designed in a manner much similar to the source documents.

Appropriate messages are provided in the message area, which prompts the user in entering the right data. Erroneous data inputs are checked at the end of each screen entry.

Data Validation

The input data is validated to minimize errors in data entry. For certain data specific codes have been given and validation is done which enables the user to enter the required data or correct them if they entered wrong codes. It uses both server side and client side validations. Story Narrator is implemented in ASP.NET 3-tier architecture, which contains Business Administration Logie Layer as middle tier the server side validation in it.

4.2 Output Design

computer output is the most important and direct source of information to the user. efficient intelligible output design improves the system relationship with the user and helps in decision making. one of the most important features of the system for users is the output it produces. output design should improve the system relationship with the user it produces and helps in decision making. considering the future use of the output required and depending on the nature, they are displayed on the monitor for immediate need or obtaining the hard copy the objective of output design is to define the control and formats of all printed, documented, reports and screens that will be produced by the system. computer output is the most important and direct source of information to the user. for many end users output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the application. output generally

refers to the results that are generated by the system. the output of the system is designed so as to include a number of reports. reports reflect the output design. objective of output design are:

• design output to serve the intended purpose.

• design output to fit the user and choose the right output method.

• deliver the appropriate quantity of output and provide output on time.

4.3 Module Description

There are 4 modules in the system

1. Admin/ University
2. College
3. Students
4. Parents

Admin /University Module

- Login
- Manage college
- View Courses
- View subiects
- View lecture notes
- View students
- Manage notifications Manage exam details
- Publish result date
- View complaints
- View reply
- View enquiry
- Send reply

College Module

- Login
- Manage departments
- Manage courses
- Manage subjects
- Add lecture notes
- Manage fees structure
- Manage students
- View parent
- message
- View notifications View exam details
- Published result date
- Send enquiry
- view reply

Students Module

- Login
- Update profile
- View departments
- View courses
- View subiect
- View lecture notes
- View fee structure View exam details
- View published result date
- View notifications
- Send complaint
- View reply

Parents Module

- Register
- Login

- Update profile
- View notifications
- Send message
- View College
- View departments
- View courses
- View exam details
- View published result date

4.4 TABLES

LOGIN

FIELD	DATA TYPE	CONSTAINTS	DESCRIPTION
Login_id	Int	Primary key	Login id
Username	Varchar	Not null	Username
Password	Varchar	Not null	Password
User type	Varchar	Not null	User type

DEPARTMENTS

FIELD	DATA TYPE	CONSTRAINTS	DESCRIPTION
Department_id	Int	Primary key	Department id
College_id	Int	Not null	College id
Department name	Varchar	Not null	Department name
description	Varchar	Not nul	Description

COLLEGE

FIELDS	DATA TYPE	CONTRAINT	DESCRIPTION
College_id	Int	Primary key	College_id
Login_id	Int	Not null	Login_id
First name	varchar	Not null	First name
Last name	Varchar	Not null	Last name
Phone	Varchar	Not null	Phone
E mail	Varchar	Not null	E mail
Place	varchar	Not null	Place
District	Varchar	Not null	District
Pincode	varchar	Not null	Pincode

STUDENTS

FIELDS	DATA TYPE	CONSTRAINT	DESCRIPTION
Students_id	Int	Primary key	Student id
Login_id	Int	Not null	Login id
College_id	Int	Not null	College id
Course_id	Int	Not null	Course id
First_name	Varchar	Not null	First name
Last_name	Varchar	Not null	Last name
Gender	Varchar	Not null	Gender
Dob	Varchar	Not null	Dob
House_name	Varchar	Not null	House name
place	Varchar	Not null	Place
pincode	Varchar	Not null	Pincode
district	Varchar	Not null	District

COMPLAINTS

FIELD	DATA TYPE	CONSTRAINTS	DESCRIPTION
Complaint_id	Int	Primary key	Complaint id
Student_id	Int	Not null	Student id
Decription	Varchar	Not null	Decription
Replay	Varchar	Not null	Replay
Complaint date	Varchar	Not null	Complaint date

ENQUIRY

Field	DATA TYPE	CONSTRAINT	DESCRIPTION
Enquiry_id	Int	Primary key	Enquiry _id
College_id	Int	Not null	College_id
Enquiry	Varchar	Not null	Enquiry
Replay	Varchar	Not null	Replay
Date	varchar	Not null	Date

COURSE

FIELDS	DATA TYPE	CONSTRAINT	DESCRIPTION
Course_id	Int	Primay key	Course_id
Department_id	int	Not null	Department_id
College_id	Int	Not null	College_id
Course	Varchar	Not null	course

FEE STRUCTURE

FIELDS	DATA TYPE	CONSTRAINT	DESCRIPTION
Fee_id	Int	Primary key	Fee_id
Course_id	int	Not null	Course_id
College_id	Int	Not null	College_id
Fees	Varchar	Not null	Fees
Details	varchar	Not null	Details

NOTIFICATION

FIELDS	DATA TYPE	CONTRIANT	DESCRIPTION
Notification _id	Int	Primary key	Notification_id
Notification	Varchar	Not null	Notification
Date	varchar	Not null	Date

EXAM DETAILS

FIELDS	DATA TYPE	CONSTRAINT	DESCRIPTION
Exam_id	Int	Primary key	Exam_id
Course_id	Int	Not null	Course_id
Exam	Varchar	Not null	Exam
Details	Varchar	Not null	Details
Date	varchar	Not null	Date

PUBLISH

FIELDS	DATA TYPE	CONSTRAINT	DESCRIPTION
Publish_id	Int	Primary key	Publish_id
Exam_id	Int	Not null	Exam_id
Published date	Varchar	Not null	Published date
Details	Varchar	Not null	Details
Date	Varchar	Not null	Date
status	varchar	Not null	Status

SUBJECT

FIELDS	DATA TYPE	CONSTRAINTS	DESCRIPTION
Subject_id	Int	Primary key	Subject_id
course _id	Int	Not null	Course_id
subjects	varchar	Not null	Subjects

LECTURE NOTES

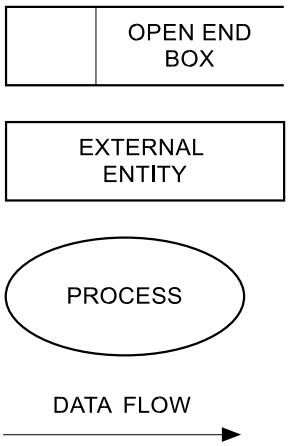
FIELDS	DATA TYPE	CONSTRAINT	DESCRIPTION
Note_id	Int	Primary key	Note_id
Subject_id	Int	Not null	Subject_id
Upload lecture note	Varchar	Not null	Upload lecture note
Date	varchar	Not null	Date

MESSAGE

FIELD	DATA TYPE	CONSTRAINT	DESCRIPTION
Message_id	Int	Primary key	Message id
Sender_id	Int	Not null	Sender id
Receiver_id	Int	Not null	Receiver id
Message	Varchar	Not null	Message
Date	Varchar	Not null	Date

4.4 DATA FLOW DIAGRAM SYMBOLS

To construct a data flow diagram the following symbols are used:



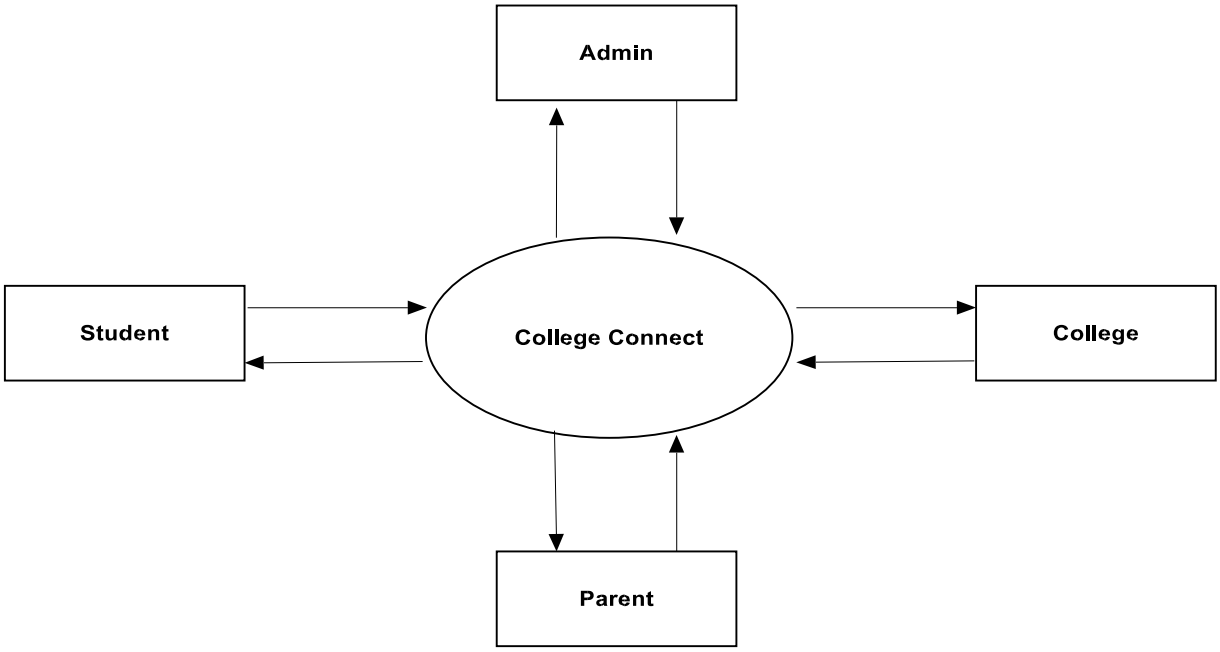
An arrow identifies the data flow in motion. It is a pipeline through which information is flow like the rectangle in the flowchart. A circle stands for process that converts data into information's. An open-ended box represents a data at rest or a temporary repository of data. A square defines a source or destinations of system data.

Five rules for constructing a data flow diagram

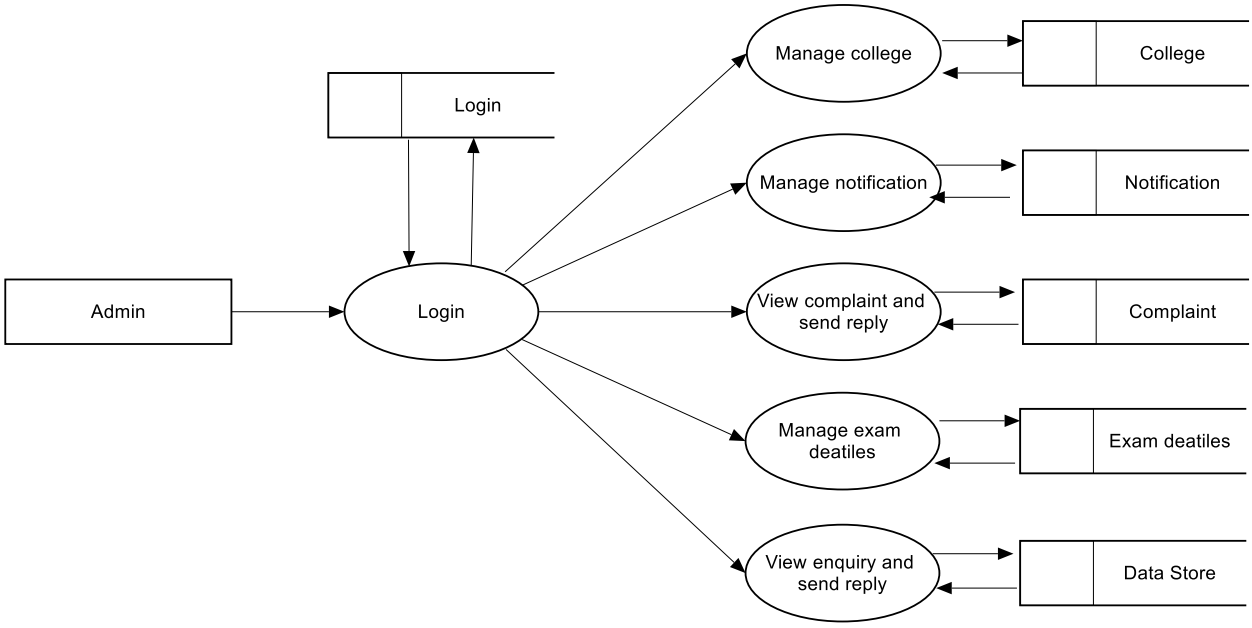
- Squares, circles and files must bear names.
- Arrow should not cross each other.
- Decomposed data flow squares and circles have same names
- Choose meaningful names for data flow
- Draw all data flows around the outside of the diagram.

DFD DIAGRAM

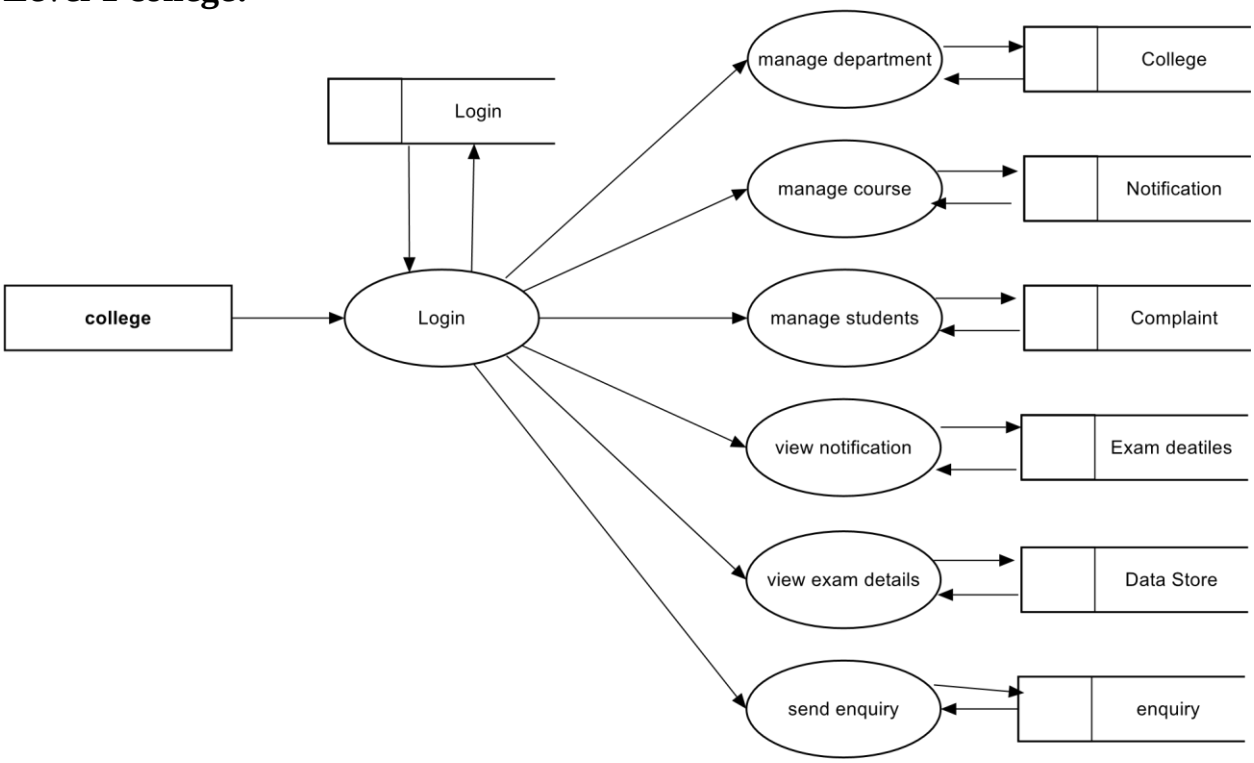
Level 0:



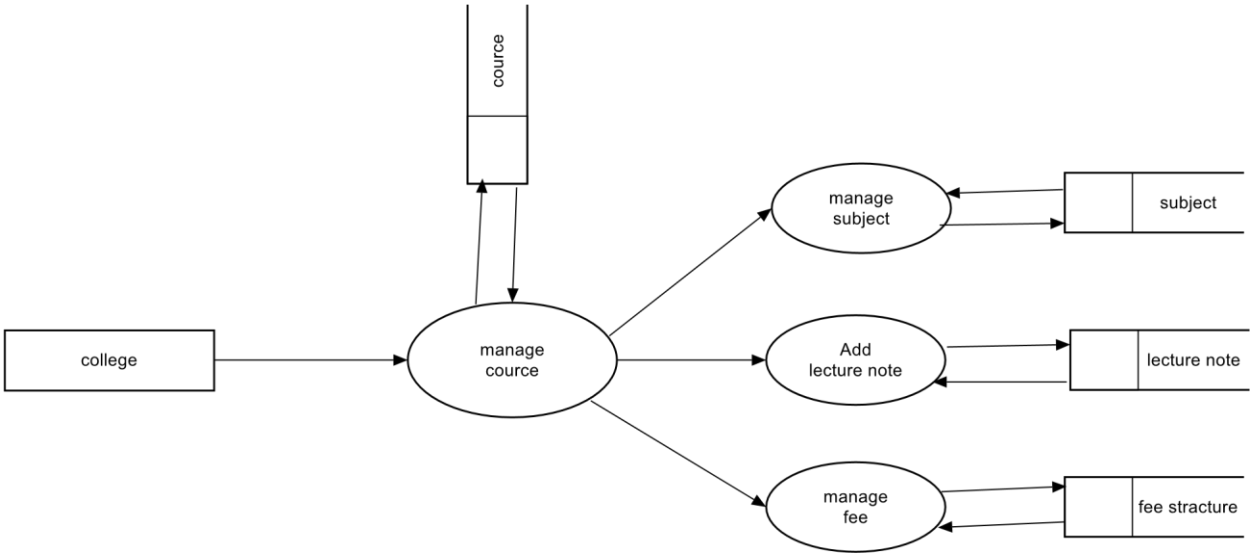
Level 1:


Login
Manage college
College
Manage notification
Notification
Admin
Login
View complaint and send reply
Complaint
Manage exam deatiles
Exam deatiles
View enquiry and send reply
Data Store

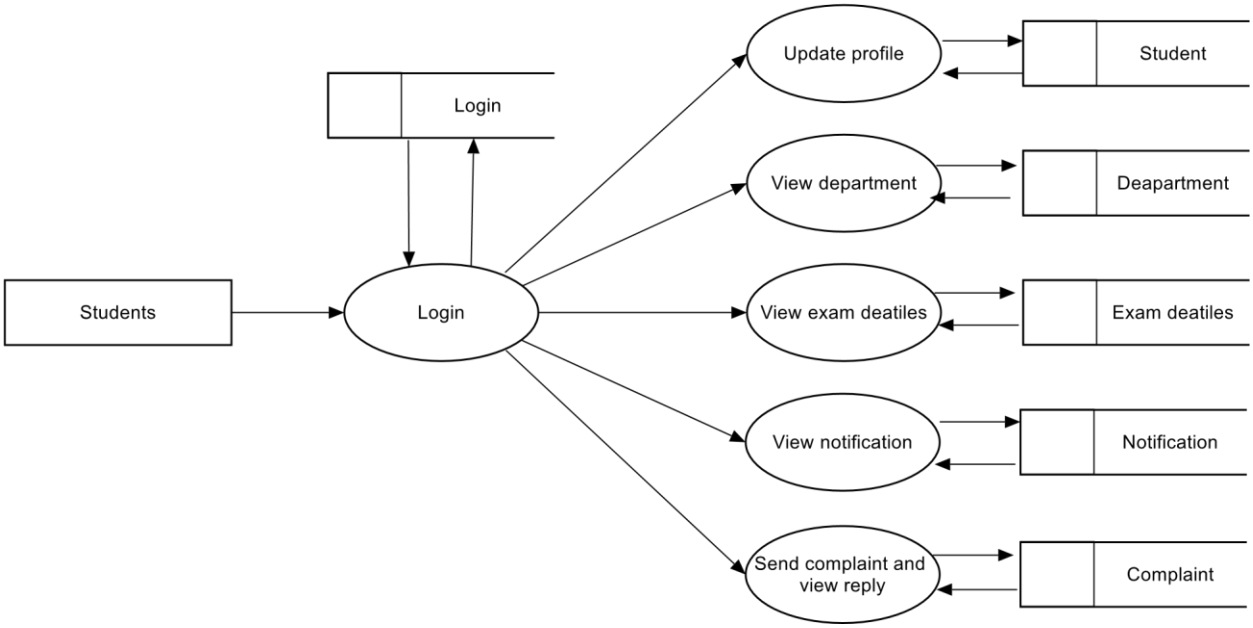
Level 1 college:



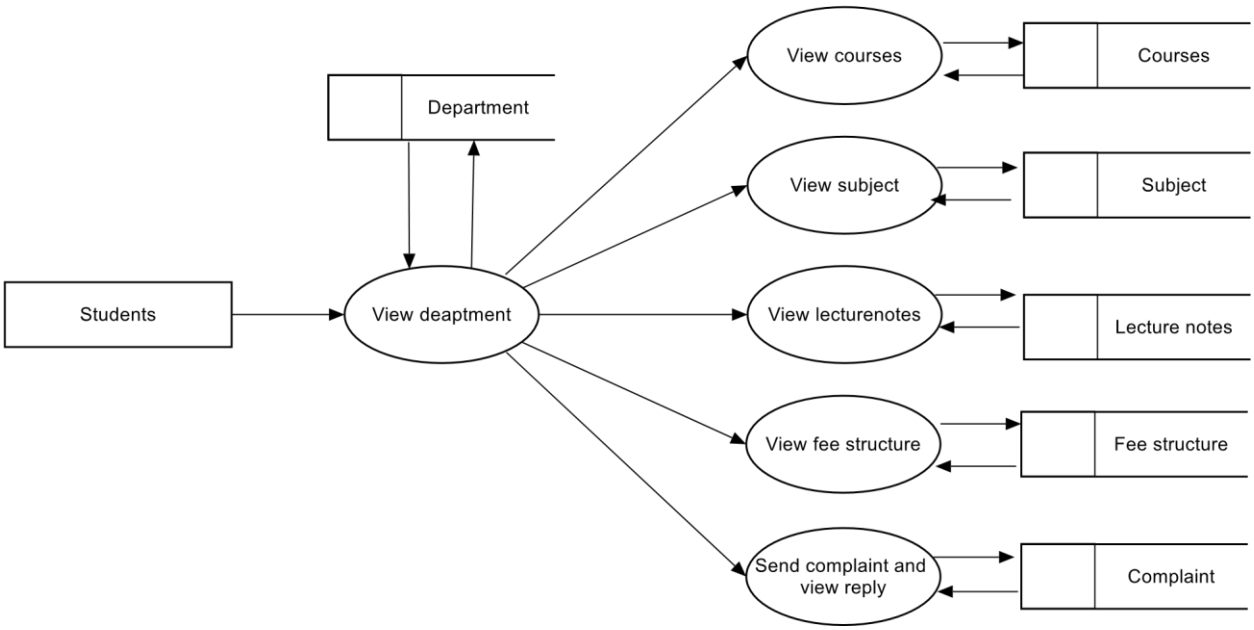
Level 2 College:



Level 1 Student:



Level 2 Student:



4.5 Entity Relationship Diagram

The overall logical structure of a database can be expressed graphically by an E-R diagram. While drawing the E-R diagram, entity names are responsible by a rectangle, relationships are represented by a diamond and oval shapes are used for representing attributes. Entity Relationship Diagram Notations

Entity:

An entity is a person, place, thing, or event of interest to the organization and about which data are captured, stored, proceed. For example, an employee is an entity.

Attribute:

Various type of data item that describe an entity are known as attributes. For example name, address, DOB(Date Of Birth)etc are the attribute of the entity employee.

Key attribute:

Key attribute is the unique, distinguishing characteristics of entity.

One-to-one (1:1)

One-to-many (1: M)

Many-to-many (M:M)

A one-to-one (I:l) relationship is an association only between two entities. A one to-many (I:M) relationship exists when one entity is related to more than one entity. A many-to many (M:M) relationship exists between more than ane entities. A database which conforms to an E-R diagram can be represented by a table or a set of tables. For each entity set and for each relationship set in the database, there is a unique table which is assigned the name of corresponding entity set or relationship set. Each table has a number of columns which are given unique names.

There are two types of entities, namely dependent entities (also called weak entities) and independent entities (also called regular or strong entities). The weak entity is the one whose existence depends on another entity. An entity set is called weak entity set if it's existence depends on the existence of another weak entity or any other strong entity. A weak entity set does not have sufficient attributes to form a primary key.

Tips to draw E-R Diagram

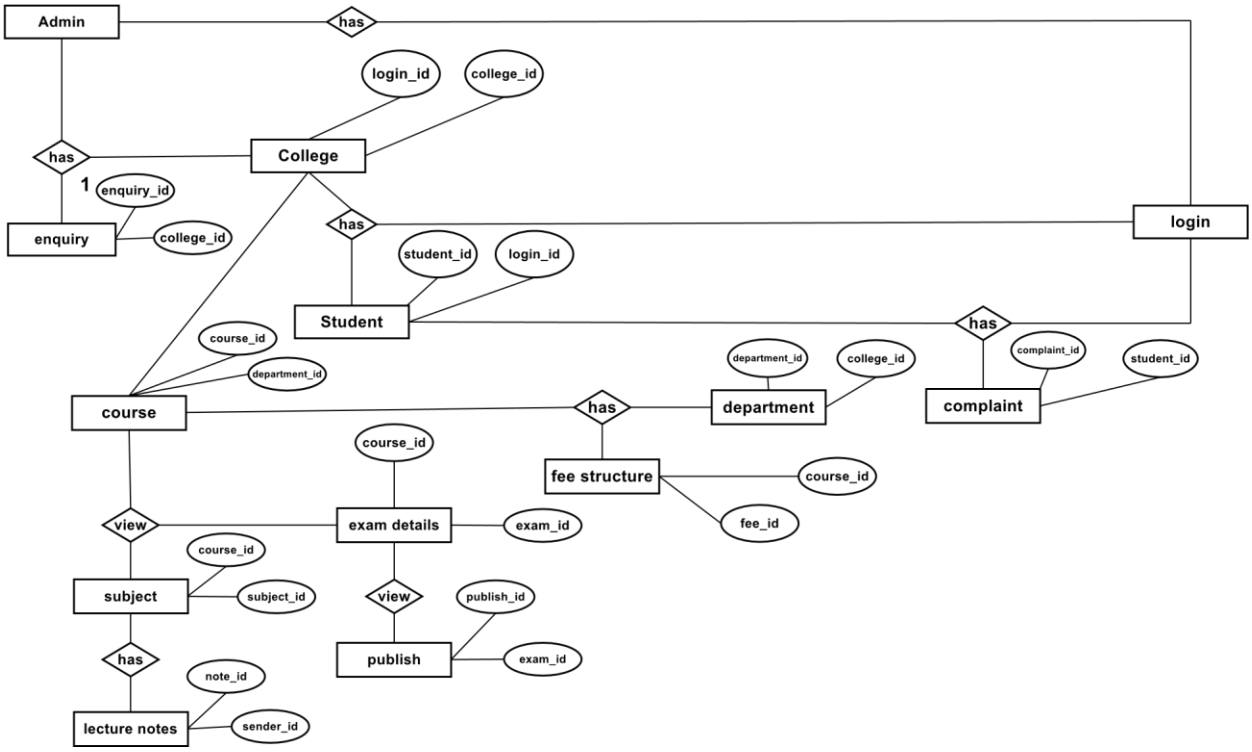
While designing database using E-R diagrams you should use the following guidelines:

• Do not introduce unwanted attributes

• Merge the entities with common attributes or purposes.

• Decompose complex entities by dividing attributes into sub-attributes.

ER DIAGRAM



FEATURES OF LANGUAGE

5.1 FEATURES OF LANGUAGE

1. ANDROID SDK

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in Java programming language using the Android Software Development Kit (SDK), but other development environments are also available. The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS 10.5.8 or later, and Windows XP OF later. The SDK is not currently available on Android, however, the software can be developed by using specialized Android application

Until around the end of 2014, the officially supported integrated development environment (IDE) was Eclipse using the AndroidDevelopment Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. As of 2015,

Android Studio, made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing. Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK

FEATURES OF ANDROID

. Near Field Communication (NEC):Most Android devices support NEC, which allows electronic devices to easily interact across short distances. The main aim here is to create a payment option that is simpler than carrying credit cards or cash, and while the market hasn't exploded as many experts had predicted, there may be an alternative in the works, in the form of Bluetooth Low Energy (BLE).

- Infrared Transmission: The Android operating system supports a built-in infrared transmitter, allowing you to use your phone or tablet as a remote control.

- Automation: The Taker app lets you not only control app permissions but also automate them.
- Alternate Keyboards: Android supports multiple keyboards and makes them easy to install; the SwiftKey, Skype, and Spen apps all offer ways to quickly change up your keyboard style. Other mobile operating systems either don't permit extra keyboards at all, or the process to install and use them are tedious and time-consuming.

- Storage and Battery Swap: Android phones also have unique hardware capabilities. Google's OS makes it possible to remove and upgrade your battery or to replace one that no longer holds a charge. In addition, Android phones come with SD card slots for expandable storage.

- Custom Home Screens: While it's possible to hack certain phones to customize the home screen, Android comes with this capability from the get-go. Download a third-party launcher like Nova, Apex or Slide and you can add gestures, new shortcuts, or even performance enhancements for older-model devices.

FRONT END TOOLS: ANDROID - JAVA

Android

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras,

PCs and other electronics. One of the most widely used mobile OS these days is ANDROID. Android is a software bunch comprising not only operating system but also middleware and key applications.Google's Android division certainly has a sense of humor: It named all of its version codenames after

JAVA

Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, runanywhere language. A platform is the hardware or software environment in which a program runs. Java is one of the world's most important and widelv used computer languages, and it has held this distinction for many years.Object Oriented meaning the capability to reuse code.It is possible to develop a single application which can run on multiple platforms like Windows, UNIX, and Macintosh systems.Java does not support the use of pointers. It automatically manages memory garbage-collection routine is activated when system runs short of memory. Java provides the most secure programming environment. Java doesn't just fix security loopholes-it eliminates them, which makes Java the perfect language for programming on the Web.

PHP

PHP was at first created as a simple scripting platform called "Personal Home Page".

Nowadays PHP is an alternative of the Microsoft's Active Server Pages (ASP) technology PHP is an open source server-side language which is used for creating dynamic web pages. It can be embedded into HTML. PHP is usually used in conjunction with a MySQL database on Linux/UNIX web servers. It is probably the most popular scripting language.

PHP is a widely-used general-purpose scripting language and interpreter that is freely available. A full explanation of all the PHP functions, complete user manual and lots of tutorials can be found on the PH's official page.PHP code may be executed with a command line interface (CLI), embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks.

FEATURES OF PHP

- Interpreted : It is an interpreted language, i.e. there is no need for compilation.
- Faster :It is faster than other scripting language e.g. asp and jsp.
- Open Source :Open source means you no need to pay for use php, you can free download and use.
- Platform Independent : PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows.
- Case Sensitive :PHP is case sensitive scripting language at time of variable declaration. In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.
- Error Reporting :PHP have some predefined error reporting constants to generate a warning or error notice.

Real-Time Access Monitoring :PHP provides access logging by creating the summary of recent accesses for the user.

5.2 Overview of packages

MySQL Database

MySQL is a fast easy- to- use RDBMS being used for many small and big businesses.

MySQL is developed, supported, and marketed by MySOL LAB, which is a Swedish company.MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SOL). SOL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

- MySQL is released under an open -source license. So you have nothing to pay

for use it.

- MySQL is very powerful program in its own right. It handles a large subset of functionality of the most expensive and powerful database packages.

- MySOL work very quickly and works well even with large data sets.

- MySQL works on any operating systems and within many languages including

JAVA, PHP, C, C++, Pearl etc.

- MySOL uses the standard form of the well-known SQL data language.

5.3 CODING

Building the refers to the coding step in software engineering process. All the software engineering steps that have been presented up to this step are directed towards a final obiectives, ie, to translate representations of software into a form that can be understood by the computer.Coding is a process that transforms designs into a programming language When considered as a step in software engineering process coding is viewed as a natural consequence of desing. However, programming language characteristics and coding style can profoundly affect the software quality and maintainability. The coding step translates a detailed representation of software into a programming language realization. The translation process

begins when a compiler accepts the source code as input and produces machine dependent object code as output. Compiler output is further is translated into machine code. Language characteristics have an impact on the quality and efficiency of translation.

Main.py

```
from flask import Flask

from admin import admin

from public import public

from colleges import

colleges from api import

api



app= Flask(__name__)


app.secret_key='aspire'


app.register_blueprint(public)

app.register_blueprint(colleges)

app.register_blueprint(admin)

app.register_blueprint(api,url_prefix='/api')
```

```
app.run(debug=True,port=5009,host="0
.0.0.0") from flask import * from
database import *


public=Blueprint('public',__name__)


@public.route('/',methods=['get','po
st']) def lander():
    return render_template('index.html')


@public.route('/login',methods=['get','p
ost']) def login():


    if "submit" in request.form:
un=request.form['uname']
pw=request.form['pswd']         qs="select *
from login where username='%s' and
password='%s' "%(un,pw)


res=select(qs)
if res:
```

```
        session['login_id']=res[0]['login_id']


        if res[0]['usertype']=="admin":
            return
redirect(url_for('admin.admin_home'))
elif res[0]['usertype']=="college":
            q="select * from college where
login_id='%s'"%(session['login_id'])            res=select(q)
if res:
                session['college_id']=res[0]['college_id']
return redirect(url_for('colleges.colleges_home'))
return render_template('login.html')


@public.route('/college_registeration',methods=['get',
'post']) def college_registeration():


    if "register" in request.form:
fn=request.form['fname']
ln=request.form['lname']
ph=request.form['pho']
em=request.form['email']
```

```
pl=request.form['place']
dst=request.form['district']
pd=request.form['pcode']
un=request.form['uname']
pw=request.form['pswd']


    ql="insert into login values(null,'%s','%s','college')"%(un,pw)
log=insert(ql)


    qr="insert into college
values(null,'%s','%s','%s','%s','%s','%s','%s','%s')"%(log,fn,ln,ph,em,pl,dst,pd)
insert(qr)      return redirect(url_for('public.login'))     return
render_template('college_registration.html')



@public.route('/logout'
) def logout():
   return render_template('index.html')
USE `college_app_aspire`;

/*Table structure for table `college` */
```

```
DROP TABLE IF EXISTS `college`;

CREATE TABLE `college` (
  `college_id` int(11) NOT NULL AUTO_INCREMENT,
  `login_id` int(11) DEFAULT NULL,
  `first_name` varchar(100) DEFAULT NULL,
  `last_name` varchar(100) DEFAULT NULL,
  `phone` varchar(100) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `place` varchar(100) DEFAULT NULL,
  `district` varchar(100) DEFAULT NULL,
  `pincode` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`college_id`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

/*Data for the table `college` */

insert  into
`college`(`college_id`,`login_id`,`first_name`,`last_name`,`phone`,`email`,`place`,`district`,`pinc
ode`) values
(1,2,'aspire','aspire','9061442294','aspire@gmail.com','pala','kottayam','686574');
```

```
/*Table structure for table `complaints` */

DROP TABLE IF EXISTS `complaints`;

CREATE TABLE `complaints` (
  `complaint_id` int(11) NOT NULL AUTO_INCREMENT,
  `student_id` int(11) DEFAULT NULL,
  `description` varchar(100) DEFAULT NULL,
  `reply` varchar(100) DEFAULT NULL,
  `complaint_date` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`complaint_id`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

/*Data for the table `complaints` */

insert  into `complaints`(`complaint_id`,`student_id`,`description`,`reply`,`complaint_date`)
values
(1,1,'Tiger Ka Hukkum','Jailer....','27-10-23');

/*Table structure for table `course` */

DROP TABLE IF EXISTS `course`;
```

```
CREATE TABLE `course` (
  `course_id` int(11) NOT NULL AUTO_INCREMENT,
  `department_id` int(11) DEFAULT NULL,
  `college_id` int(11) DEFAULT NULL,
  `course` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`course_id`)
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

/*Data for the table `course` */

insert  into `course`(`course_id`,`department_id`,`college_id`,`course`) values
(7,2,2,'BA malayalam'),
(2,1,2,'computer Application'),
(5,1,2,'Chattered Accountant');

/*Table structure for table `departments` */

DROP TABLE IF EXISTS `departments`;

CREATE TABLE `departments` (
  `department_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `college_id` int(11) DEFAULT NULL,
  `department_name` varchar(100) DEFAULT NULL,

  `description` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`department_id`)

) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;


/*Data for the table `departments` */


insert  into `departments`(`department_id`,`college_id`,`department_name`,`description`) values

(1,2,'B.Com','Bachelor Of Comerce'),

(2,2,'BA','Bachelor of Arts');


/*Table structure for table `enquiry` */


DROP TABLE IF EXISTS `enquiry`;


CREATE TABLE `enquiry` (

  `enquiry_id` int(11) NOT NULL AUTO_INCREMENT,

  `college_id` int(11) DEFAULT NULL,

  `enquiry` varchar(100) DEFAULT NULL,

  `reply` varchar(100) DEFAULT NULL,

  `enquiry_date` varchar(100) DEFAULT NULL,
```

```
  PRIMARY KEY (`enquiry_id`)

) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;


/*Data for the table `enquiry` */


insert  into `enquiry`(`enquiry_id`,`college_id`,`enquiry`,`reply`,`enquiry_date`) values

(1,2,'aysheriiiiii.....','hammeeeee.....','2023-09-06 13:40:22');
/*Table structure for table `exam_details` */


DROP TABLE IF EXISTS `exam_details`;


CREATE TABLE `exam_details` (

  `exam_id` int(11) NOT NULL AUTO_INCREMENT,

  `course_id` int(11) DEFAULT NULL,

  `exam` varchar(100) DEFAULT NULL,

  `details` varchar(100) DEFAULT NULL,

  `exam_date` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`exam_id`)

) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;


/*Data for the table `exam_details` */
```

```
insert  into `exam_details`(`exam_id`,`course_id`,`exam`,`details`,`exam_date`) values

(4,5,'maths','haiiiiii.....','2023-09-07'),

(3,7,'Malayalam','heeee heeeee hii ','2024-10-22');


/*Table structure for table `fee_structure` */


DROP TABLE IF EXISTS `fee_structure`;


CREATE TABLE `fee_structure` (

  `fee_id` int(11) NOT NULL AUTO_INCREMENT,

  `course_id` int(11) DEFAULT NULL,

  `college_id` int(11) DEFAULT NULL,
  `fees` varchar(100) DEFAULT NULL,

  `details` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`fee_id`)

) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;


/*Data for the table `fee_structure` */


insert  into `fee_structure`(`fee_id`,`course_id`,`college_id`,`fees`,`details`) values

(2,7,2,'200000','for 3 yrs'),

(3,2,2,'3000000','for 5 yrs'),
```

```
(4,5,2,'40000000','for 3 yrs');

/*Table structure for table `lecture_notes` */

DROP TABLE IF EXISTS `lecture_notes`;

CREATE TABLE `lecture_notes` (
  `note_id` int(11) NOT NULL AUTO_INCREMENT,
  `subject_id` int(11) DEFAULT NULL,
  `upload_note` varchar(1000) DEFAULT NULL,
  `date` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`note_id`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

/*Data for the table `lecture_notes` */

insert  into `lecture_notes`(`note_id`,`subject_id`,`upload_note`,`date`) values
(1,1,'static/notes9459d25c-0adc-4904-96e4-48b2cf625d66java script notes.docx','2023-09-05
10:12:43'); /*Table structure for table `login` */

DROP TABLE IF EXISTS `login`;
```

```
CREATE TABLE `login` (
  `login_id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(100) DEFAULT NULL,
  `password` varchar(100) DEFAULT NULL,
  `usertype` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`login_id`)
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

/*Data for the table `login` */

insert  into `login`(`login_id`,`username`,`password`,`usertype`) values
(1,'admin','admin','admin'),
(2,'aspire@1','aspire@1','college'),
(7,'Arun@1','Arun@1','student'),
(4,'anandhu@1','anandhu@1','student');

/*Table structure for table `message` */

DROP TABLE IF EXISTS `message`;

CREATE TABLE `message` (
```

```
  `message_id` int(11) NOT NULL AUTO_INCREMENT,

  `sender_id` int(11) DEFAULT NULL,
  `reciever_id` int(11) DEFAULT NULL,

  `message` varchar(100) DEFAULT NULL,

  `message_date` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`message_id`)

) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;



/*Data for the table `message` */



insert  into `message`(`message_id`,`sender_id`,`reciever_id`,`message`,`message_date`) values

(1,2,1,'halooooo','2023-09-07'),

(2,2,1,'halooooo','2023-09-07'),

(3,1,2,'hai','2023-09-08');



/*Table structure for table `notifications` */



DROP TABLE IF EXISTS `notifications`;



CREATE TABLE `notifications` (

  `notification_id` int(11) NOT NULL AUTO_INCREMENT,

  `notification` varchar(100) DEFAULT NULL,
```

```
  `notification_date` varchar(50) DEFAULT NULL,

  PRIMARY KEY (`notification_id`)

) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;


/*Data for the table `notifications` */


insert  into `notifications`(`notification_id`,`notification`,`notification_date`) values

(3,'checkinggg.....','2023-09-06 10:03:22');
/*Table structure for table `parent` */


DROP TABLE IF EXISTS `parent`;


CREATE TABLE `parent` (

  `parent_id` int(11) NOT NULL AUTO_INCREMENT,

  `login_id` int(11) DEFAULT NULL,

  `student_id` int(11) DEFAULT NULL,

  `firstname` varchar(100) DEFAULT NULL,

  `lastname` varchar(100) DEFAULT NULL,

  `housename` varchar(100) DEFAULT NULL,

  `mail` varchar(100) DEFAULT NULL,

  `contact` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`parent_id`)
```

```
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;


/*Data for the table `parent` */


insert  into
`parent`(`parent_id`,`login_id`,`student_id`,`firstname`,`lastname`,`housename`,`mail`,`contact`)
values  (1,4,6,'AAAAAA','BBBBB','CCCCCC','DDDDD','123456789');


/*Table structure for table `publish` */


DROP TABLE IF EXISTS `publish`;


CREATE TABLE `publish` (
  `publish_id` int(11) NOT NULL AUTO_INCREMENT,
  `exam_id` int(11) DEFAULT NULL,
  `published_date` varchar(100) DEFAULT NULL,
  `details` varchar(100) DEFAULT NULL,
  `date` varchar(100) DEFAULT NULL,
  `status` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`publish_id`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

```
/*Data for the table `publish` */

insert  into `publish`(`publish_id`,`exam_id`,`published_date`,`details`,`date`,`status`) values
(1,4,'2023-09-15','result','2023-09-06','pending');

/*Table structure for table `students` */

DROP TABLE IF EXISTS `students`;

CREATE TABLE `students` (
  `student_id` int(11) NOT NULL AUTO_INCREMENT,
  `login_id` int(11) DEFAULT NULL,
  `college_id` int(11) DEFAULT NULL,
  `course_id` int(11) DEFAULT NULL,
  `first_name` varchar(100) DEFAULT NULL,
  `last_name` varchar(100) DEFAULT NULL,
  `gender` varchar(100) DEFAULT NULL,
  `dob` varchar(100) DEFAULT NULL,
  `house_name` varchar(100) DEFAULT NULL,
  `place` varchar(100) DEFAULT NULL,
  `pincode` varchar(100) DEFAULT NULL,
  `district` varchar(100) DEFAULT NULL,
```

```
  PRIMARY KEY (`student_id`)

) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;


/*Data for the table `students` */


insert  into
`students`(`student_id`,`login_id`,`college_id`,`course_id`,`first_name`,`last_name`,`gender`,`do
b`,`house_name`,`pl ace`,`pincode`,`district`) values

(6,7,2,7,'Arun','P S','Male','1999-10-17','Parakkunnel','Pala','686573','Kottayam'),

(3,4,2,5,'Anandhu','clarion','Male','1980-02-13','medicaltrust','chalakkudi','686510','thrissur');


/*Table structure for table `subject` */


DROP TABLE IF EXISTS `subject`;


CREATE TABLE `subject` (

  `subject_id` int(11) NOT NULL AUTO_INCREMENT,

  `course_id` int(11) DEFAULT NULL,

  `subjects` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`subject_id`)

) ENGINE=MyISAM AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;
```

```
/*Data for the table `subject` */


insert  into `subject`(`subject_id`,`course_id`,`subjects`) values

(12,4,'english'),
(8,2,'computer'),

(5,3,'archaeology'),

(11,5,'Maths'),


(13,7,'malayalam');


/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

College.py

```
from flask import *
from database
import * import
uuid



colleges=Blueprint('colleges',__name__)


@colleges.route('/colleges_hom
e',) def colleges_home():
    return render_template('college_home.html')



@colleges.route('/college_add_department',methods=['get
','post']) def college_add_department():

    data={}     if "ADD" in request.form:         dpt=request.form['dptname']
des=request.form['desc']         qdpt="insert into departments
values(null,'%s','%s','%s')"%(session['login_id'],dpt,des)         insert(qdpt)
return redirect(url_for('collegescollege_add_department'))
```

```
    qvd="select * from departments where
college_id='%s'"%(session['login_id'])
data['dptmview']=select(qvd)     return
render_template('college_add_department.html',data=data)


@colleges.route('/college_add_course',methods=['get'
,'post']) def college_add_course():
    data={}     qdp="select * from departments"
data['dptview']=select(qdp)     if "ADDC" in request.form:
d=request.form['dptid']         c=request.form['crs']         qcrs="insert
into course values(null,'%s','%s','%s')"%(d,session['login_id'],c)
insert(qcrs)


    if'action' in request.args:
        action=request.args['action']
crsid=request.args['crsid']


else:
        action=None


    if action=='delete':
```

```
        qdel="delete        from        course        where
course_id='%s'"%(crsid)       delete(qdel)       return
redirect(url_for('colleges.college_add_course'))


    if action=='update':
        qsc="select * from course inner join departments using (department_id) where
course_id='%s'"%(crsid)        data['cosview']=select(qsc)


    if "update" in request.form:        dp=request.form['dptid']        cr=request.form['crs']
qcrs="update  course set
department_id='%s',college_id='%s',course='%s'"%(dp,session['login_id'],cr)
        insert(qcrs)        return
redirect(url_for('colleges.college_add_course'))


    qcurs="SELECT * ,CONCAT (college.first_name,' ',college.last_name) AS college_name
FROM course INNER JOIN departments USING(department_id) INNER JOIN college ON
course.college_id=college.login_id where login_id='%s'"%(session['login_id'])
    data['cursview']=select(qcurs)     return
render_template('college_add_course.html',data=dat
a)
```

```
@colleges.route('/college_add_subjects',methods=['get'
,'post']) def college_add_subjects():

   data={}     qcs="SELECT * FROM course WHERE
college_id='%s'"%(session['login_id'])     data['csview']=select(qcs)


   if "ADDS" in request.form:
c=request.form['csid']         s=request.form['sbjt']
qsbt="insert into `subject`
values(null,'%s','%s')"%(c,s)         insert(qsbt)
print(qsbt)


   if'action' in request.args:
      action=request.args['action']
cursid=request.args['cursid']


else:
      action=None


   if action=='delete':
      qel="delete from `subject` where course_id='%s'"%(cursid)
```

```
    delete(qel)        return
redirect(url_for('colleges.college_add_subjects'))
if action=='update':

    qsc="select * from subject inner join course using (course_id) where
subject_id='%s'"%(cursid)        data['sjtview']=select(qsc)


  if "update" in request.form:        crs=request.form['csid']
sb=request.form['sbjt']         qsb="update `subject` set course_id='%s',subjects='%s'
where subject_id='%s' "%(crs,sb,cursid)        update(qsb)



  qsbts="SELECT * FROM `subject` INNER JOIN course USING
(course_id) WHERE college_id='%s'"%(session['login_id'])
data['sbtview']=select(qsbts)    return
render_template('college_add_subjects.html',data=data)


@colleges.route('/college_add_notes',methods=['get','post']) def
college_add_notes():     if "upload" in request.form:
s=request.args['subid']        notes=request.files['pdf']
path="static/notes"+str(uuid.uuid4())+notes.filename
notes.save(path)        qno="insert into `lecture_notes`
values(null,'%s','%s',now())"%(s,path)        insert(qno)
return redirect(url_for('colleges.college_add_subjects'))
```

```
return render_template('college_add_notes.html')

@colleges.route('/college_add_fees',methods=['get','post']) def

college_add_fees():



   data={}     if "submit"

in request.form:

cr=request.args['crsid']



       fe=request.form['fee']

dl=request.form['dtls']



       qfe="insert into `fee_structure`

values(null,'%s','%s','%s','%s')"%(cr,session['login_id'],fe,dl)         insert(qfe)

return redirect(url_for('colleges.college_add_course'))



   qfee="SELECT *,CONCAT(college.first_name,' ',college.last_name)AS college_name
FROM `fee_structure`
INNER JOIN course USING (course_id) INNER JOIN college

ON(college.login_id=`fee_structure`.college_id) WHERE

`fee_structure`.college_id='%s'"%(session['login_id'])     data['feview']=select(qfee)



   if "action" in

request.args:
```

```
action=request.args['action']
fid=request.args['fid']
else:
        action=None

    if action=='delete':
        qfd="delete from `fee_structure` where
fee_id='%s'"%(fid)          delete(qfd)
       return redirect(url_for('colleges.college_add_course'))

    if action=='update':
        qf="select * from `fee_structure` where fee_id='%s'"%(fid)
data['feesview']=select(qf)     if "update" in request.form:
f=request.form['fee']          d=request.form['dtls']          qfu="update
`fee_structure` set fees='%s', details='%s' where fee_id='%s'"%(f,d,fid)
update(qfu)          return redirect(url_for('colleges.college_add_course'))

    return render_template('college_add_fees.html',data=data)

@colleges.route('/college_add_students',methods=['pos
t','get']) def college_add_students():
    data={}
```

```
    qscr="select * from course where
college_id='%s'"%(session['login_id'])
data['qscrview']=select(qscr)     wrc="select * from parent"
data['viewperent']=select(wrc)


    if "submit" in request.form:
cu=request.form['curs']
fn=request.form['fname']
ln=request.form['lname']
ge=request.form['Gender']
do=request.form['db']
hn=request.form['hname']
pl=request.form['place']
pc=request.form['pcode']
di=request.form['disrict']
un=request.form['uname']
ps=request.form['pswd']
parent=request.form['parent']


        qst="insert into login values(null,'%s','%s','student')"%(un,ps)
logid=insert(qst)
```

```
        qsd="insert into students
values(null,'%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s')"%(logid,session['login_id'],c
u,fn,ln,ge,do,hn,pl,p c,di,parent)        insert(qsd)        return
redirect(url_for('colleges.college_add_students'))



    if "action" in request.args:
action=request.args['action']
sd=request.args['stid']


else:

        action=None


    if action=='delete':

        qds="delete from students where login_id='%s'"%(sd)
       delete(qds)


        qdls="delete from login where
login_id='%s'"%(sd)        delete(qdls)        return
redirect(url_for('colleges.college_add_students'))


    if action=='update':
```

```
        qss="select * from students inner join login using (login_id) where login_id='%s'"%(sd)
data['sdview']=select(qss)


    if "updateS" in request.form:


        cur=request.form['curs']
fna=request.form['fname']
lna=request.form['lname']
gen=request.form['Gender']
d=request.form['db']
hna=request.form['hname']
pla=request.form['place']
pco=request.form['pcod']
dis=request.form['disrict']
una=request.form['uname']
psw=request.form['pswd']


        qul="update login set username='%s', password='%s' where login_id='%s'
"%(una,psw,sd)        logid=insert(qul)


       qus="update students set
course_id='%s',first_name='%s',last_name='%s',gender='%s',dob='%s',house_name='%s',place='
%s',pincode='%s',di strict='%s' where login_id='%s' "%(cur,fna,lna,gen,d,hna,pla,pco,dis,sd)
```

```
    insert(qus)        return
redirect(url_for('colleges.college_add_students'))



  qstud="SELECT *,CONCAT(students.first_name,' ',students.last_name)AS name FROM
`students` INNER JOIN course USING(course_id) where
students.college_id='%s'"%(session['login_id'])    data['stview']=select(qstud)    return
render_template('college_add_students.html',data=data)


@colleges.route('/college_view_notificat
ion') def college_view_notification():

  data={}    qs="SELECT * FROM `notifications`"
data['nview']=select(qs)    return
render_template('college_view_notifications.html',data=dat
a)


@colleges.route('/college_view_p
arent') def college_view_parent():
data={}    sd=request.args['stid']


  qs="SELECT *,CONCAT(`parent`.firstname,' ',`parent`.lastname)AS
parent_name,CONCAT(`students`.first_name,' ',`students`.last_name)AS student_name FROM
`parent` INNER JOIN students USING (parent_id)  where parent_id='%s' "%(sd)
```

```
    print(qs)
data['parview']=select(qs)
return
render_template('college_view
_parent.html',data=data)


@colleges.route('/college_view_exam') def college_view_exam():     data={}
qex="SELECT * FROM `exam_details` INNER JOIN course USING(course_id)
INNER JOIN college ON(`college`.login_id=`course`.college_id) where
course.college_id='%s'"%(session['login_id'])     data['exview']=select(qex)     return
render_template('college_view_exam.html',data=data)


@colleges.route('/college_view_resultd
ate') def college_view_resultdate():

    data={}


    qrd="SELECT * FROM publish INNER JOIN `exam_details` USING(exam_id) INNER
JOIN `course` USING
(course_id) where college_id='%s'"%(session['login_id'])
data['pubview']=select(qrd)


    return render_template('college_view_resultdate.html',data=data)
```

```
@colleges.route('/college_send_enquiry',methods=['pos
t','get']) def college_send_enquiry():

    data={}


    if "submit" in request.form:
e=request.form['enq']         qin="insert into
`enquiry`
values(null,'%s','%s','pending',now())"%(se
ssion['login_id'],e)         insert(qin)


    qes="select * from `enquiry` where
college_id='%s'"%(session['login_id'])
data['enqview']=select(qes)     return
render_template('college_send_enquiry.html',data=data)


@colleges.route('/college_send_message',methods=['
get','post']) def college_send_message():     data={}
sender_id=session['login_id']
data['college']=sender_id


    if "chat" in request.form:         paid=request.args['paid']
messsage=request.form['messsage']         quc="insert into `message`
```

values(null,'%s','%s','%s',curdate())"%(sender_id,paid,messsage) insert(quc)

print(quc) return redirect(url_for('colleges.college_send_message'))

qch="SELECT * FROM `message` WHERE`sender_id`='%s' OR

`reciever_id`='%s'

"%(session['login_id'],sender_id) print(qch)

data['message']=select(qch) return

render_template('college_send_message.html',data=dat

a)

Database.py

import mysql.connector

user="root" password=""

database="college_app_as

pire"

def select(q):

con=mysql.connector.connect(user=user,password=password,host="localhost",database=

database) cur=con.cursor(dictionary=True) cur.execute(q)

result=cur.fetchall() cur.close() con.close() return result

```
def insert(q):

con=mysql.connector.connect(user=user,password=password,host="localhost",database=
database)      cur=con.cursor(dictionary=True)      cur.execute(q)          con.commit()

result=cur.lastrowid
cur.close()
con.close()    return
result

def update(q):

con=mysql.connector.connect(user=user,password=password,host="localhost",database=
database)      cur=con.cursor(dictionary=True)      cur.execute(q)          con.commit()
result=cur.rowcount   cur.close()     con.close()

def delete(q):

con=mysql.connector.connect(user=user,password=password,host="localhost",database=
database)      cur=con.cursor(dictionary=True)      cur.execute(q)          con.commit()
result=cur.rowcount   cur.close()     con.close()
```

api.py

```
from flask import *
from database
import * import
uuid


api=Blueprint('api',__name__)



@api.route("/login")
def login():
        data={}


         uname=request.args['username']
pwd=request.args['password']
qr="select * from login"
val=select(qr)

      if val[0]['usertype']=='student':

                  q="SELECT *,CONCAT(first_name,' ',last_name) AS  zname FROM login
INNER JOIN students USING(login_id)  where username='%s' and password='%s'
"%(uname,pwd)

                  print(q)
```

```
                res=select(q)

        elif val[0]['usertype']=='parent':

            q="SELECT *,CONCAT(firstname,' ',lastname) AS  zname FROM login INNER JOIN
parent
USING(login_id) inner join students using(parent_id) where username='%s' and password='%s'
"%(uname,pwd)

                print(q)

                res=select(q)

          data['status']='success'
data['data']=res



         return str(data)


@api.route("/customerreg
") def customerreg():

    data={}


    fname=request.args['fname']
lname=request.args['lname']
hname=request.args['hname']
place=request.args['place']
pin=request.args['pincode']
phone=request.args['phone']
```

```
email=request.args['email']
uname=request.args['username']
passw=request.args['password']


   q="select * from login where
username='%s'"%(uname)     res=select(q)     if
res:


data['status']='duplicate'
else:
      q="insert into `login`
values(NULL,'%s','%s','parent')"%(uname,passw)         res=insert(q)


      w="insert into parent
value(NULL,'%s','%s','%s','%s','%s','%s','%s','%s')"%(res,fname,lname,hname,place,phone,pin,e
mail)
      insert(w)
flash("Registration
Successfull")
data['status']='success'


   return str(data)
```

```
@api.route('/userviewprofile
/') def userviewprofile():
        data={} log=request.args['login_id']
        q="SELECT * from students where
        login_id='%s' "%(log)


res=select(q)   if
res:
                data['status']='success'
data['data']=res
              data['method']='userviewprofile'
        else:
                data['status']='failed'
data['method']='userviewprofile'       return
str(data)


@api.route('/Student_update_profi
le') def Student_update_profile():
        data={}
```

```
        fname =
request.args['fname']   lname =
request.args['lname']
hname=request.args['hname']
place= request.args['place']
district = request.args['district']
pin = request.args['pincode']
logid=request.args['logid']
q="UPDATE `students` SET
`first_name`='%s',`last_name`='%s',`house_name`='%s',`place`='%s',`pincode`='%s',`district`='%
s' WHERE
`login_id`='%s' "%(fname,lname,hname,place,pin,district,logid)
        update(q)
       data['status']='success'
         data['method']='userupprof'
         return str(data)


@api.route('/Student_view_departme
nt/') def Student_view_department():
        data={}
         q="SELECT * from
departments "            res=select(q)
if res:
```

```
                data['status']='success'
data['data']=res
            data['method']='Student_view_department'
        else:
                data['status']='failed'
data['method']='Student_view_department'    return
str(data)


@api.route('/Student_view_cours
e/') def Student_view_course():
        data={}
         department_id=request.args['department_id']          q="SELECT * FROM
`course` inner join departments USING(department_id)  WHERE
`department_id`='%s' "%(department_id)
         print(q)
res=select(q)   if
res:
                data['status']='success'
data['data']=res
data['method']='Student_view_course'
         else:
```

```
                data['status']='failed'
data['method']='Student_view_course'          return
str(data)




@api.route('/Student_view_fee_structu
re/') def Student_view_fee_structure():
        data={}
         course_id=request.args['course_id']   q="SELECT * FROM `fee_structure` INNER JOIN
`college` USING(`college_id`) INNER JOIN `course` USING(`course_id`) WHERE
`course_id`='%s' "%(course_id)
         print(q)
res=select(q)   if
res:
                data['status']='success'
data['data']=res
            data['method']='Student_view_fee_structure'
        else:
```

```
                data['status']='failed'
data['method']='Student_view_fee_structure'
return str(data)


@api.route('/Student_view_subje
ct/') def Student_view_subject():
        data={} course_id=request.args['course_id'] q="SELECT * FROM `subject` INNER
        JOIN `course` USING (`course_id`) WHERE `course_id`='%s'
"%(course_id)


res=select(q)   if
res:

                data['status']='success'
data['data']=res

            data['method']='Student_view_subject'

        else:

                data['status']='failed'
data['method']='Student_view_subject'         return
str(data)
```

```
@api.route('/User_download_file
s/') def User_download_files():

        data={}

         subject_id=request.args['subject_id']          q="SELECT * FROM `subject`
INNER JOIN `lecture_notes` USING (`subject_id`) WHERE
`subject_id`='%s' "%(subject_id)


res=select(q)    if
res:

                data['status']='success'
data['data']=res

            data['method']='User_download_files'

        else:

              data['status']='failed'
            data['method']='User_download_files'

         return str(data)




@api.route('/Student_view_exam_deta
ils/') def
Student_view_exam_details():
```

```
        data={}
         course_id=request.args['course_id']    q="SELECT * FROM `exam_details` inner
join course using(course_id) WHERE `course_id`='%s'
"%(course_id)

res=select(q)    if
res:
                data['status']='success'
data['data']=res
            data['method']='Student_view_exam_details'
        else:
                data['status']='failed'
data['method']='Student_view_exam_details'
return str(data)



@api.route('/Student_view_exam_date_pub
lish/') def
Student_view_exam_date_publish():
        data={}
```

```
    exam_id=request.args['exam_id']
        q="SELECT * FROM `exam_details` inner join publish using(exam_id) WHERE
        `exam_id`='%s'
"%(exam_id)



res=select(q)   if

res:

                data['status']='success'

data['data']=res

            data['method']='Student_view_exam_date_publish'

        else:

                data['status']='failed'

data['method']='Student_view_exam_date_publish'  return

str(data)









@api.route('/view_complain

t') def view_complaint():

        data={}

    log_id=request.args['log_id']
```

```
 q="SELECT * FROM `complaints` WHERE `student_id`=(SELECT `student_id` FROM
`students` WHERE `login_id`='%s')"%(log_id)

        print(q)

res=select(q)   if

res:

                data['status']="success"

data['data']=res

        else:

                data['status']="failed"

        data['method']="view_complaint"

       return str(data)


@api.route('/User_send_complai

nt') def User_send_complaint():

        data={}

        log_id=request.args['log_id']          complaint=request.args['complaint']

q="INSERT INTO `complaints` VALUES(NULL,(SELECT `student_id` FROM `students`

WHERE

`login_id`='%s'),'%s','pending',CURDATE())"%(log_id,complaint)
```

```
        print(q)
res=insert(q)    if
res:


data['status']="success"
data['data']=res          else:

                data['status']="failed"
data['method']="User_send_complaint"
return str(data)



@api.route('/Student_view_notificati
on') def Student_view_notification():

        data={}
    q="SELECT * FROM `notifications`"
        print(q)
         res=select(q)
         if res:
              data['status']="success"
                 data['data']=res
else:
```

```
                data['status']="failed"


data['method']="Student_view_notification"
return str(data)


@api.route('/viewParent_update_profi
le/') def viewParent_update_profile():
        data={}
         log=request.args['login_id']    q="SELECT *
from parent where login_id='%s' "%(log)
res=select(q)    if res:
                data['status']='success'
data['data']=res
              data['method']='userviewprofile'
        else:
                data['status']='failed'
data['method']='userviewprofile'        return
str(data)


@api.route('/Parent_update_profi
le') def Parent_update_profile():
```

```
        data={}

         fname = request.args['fname']
         lname =
request.args['lname']

hname=request.args['hname']

place= request.args['place']

district = request.args['district']

pin = request.args['pincode']

logid=request.args['logid']

q="UPDATE `parent` SET

`firstname`='%s',`lastname`='%s',`housename`='%s',`place`='%s',`pincode`='%s',`email`='%s'
WHERE
`login_id`='%s' "%(fname,lname,hname,place,pin,email,logid)

        update(q)

         data['status']='success'

data['method']='userupprof'

return str(data)









@api.route('/Parent_view_notificati

on') def Parent_view_notification():
```

```
        data={}
      q="SELECT * FROM `notifications`"
        print(q)
res=select(q)   if
res:


data['status']="success"
data['data']=res          else:
                  data['status']="failed"
        data['method']="Parent_view_notificatio
        n" return str(data)



@api.route('/Parent_view_colle
ge') def Parent_view_college():
        data={}
        q="SELECT * FROM
`college`"       print(q)
res=select(q)   if res:


data['status']="success"
data['data']=res          else:
```

```
                data['status']="failed"


data['method']="Parent_view_college"
return str(data)

@api.route('/Parent_view_depart
ment/') def
Parent_view_department():
data={}
collage_id=request.args['collage_
id'] q="SELECT * FROM
`departments` INNER JOIN
`college` USING(`college_id`)
"%(collage_id) res=select(q)

        if res:

                data['status']='success'
data['data']=res

              data['method']='Parent_view_department'

        else:

                data['status']='failed'
data['method']='Parent_view_department'     return
str(data)
```

```
@api.route('/Parent_view_cours
e/') def Parent_view_course():
        data={}
         department_id=request.args['department_id']          q="SELECT * FROM
`course` inner join departments USING(department_id)  WHERE
`department_id`='%s' "%(department_id)
         print(q)
res=select(q)   if
res:
                data['status']='success'
data['data']=res
              data['method']='Parent_view_course'
        else:
                data['status']='failed'
data['method']='Parent_view_course' return str(data)



@api.route('/Parent_view_exam_deta
ils/') def Parent_view_exam_details():
```

```
        data={}
         q="SELECT * FROM `exam_details` inner join
using(course_id)"       res=select(q)    if res:
                data['status']='success'
data['data']=res
              data['method']='Parent_view_exam_details'
         else:
                data['status']='failed'
data['method']='Parent_view_exam_details'   return
str(data)



@api.route('/Parent_view_exam_date_publ
ish/') def
Parent_view_exam_date_publish():
        data={}
         exam_id=request.args['exam_id']       q="SELECT * FROM `exam_details` inner join
publish (exam_id) WHERE `exam_id`='%s' "%(exam_id)    res=select(q)    if res:
                data['status']='success'
         data['data']=res
```

```
        data['method']='Parent_view_exam_da
        te_publish'

        else:
                data['status']='failed'
data['method']='Parent_view_exam_date_publish'    return
str(data)


@api.route('/chat')
def chat():


       data={}
        sender_id=request.args['sender_id']   receiver_id=request.args['receiver_id']
details=request.args['details']            q="insert into message
values(null,'%s','%s','%s',curdate())" %(sender_id,receiver_id,details)

        print(q)
insert(q)

        data['status']="success"
data['method']="chat"
return str(data)
```

```
@api.route('/chatdetail'
) def chatdetail():
        data={}
         sender_id=request.args['sender_id']    receiver_id=request.args['receiver_id']
q="SELECT * FROM message WHERE (sender_id='%s' AND receiver_id='%s') or
(sender_id='%s' AND receiver_id='%s')" %(sender_id,receiver_id,receiver_id,sender_id)
         print(q)
         res=select(q)
         if res:

data['status']="success"
data['data']=res          else:
                data['status']="failed"
         data['method']="chatdetail"
return str(data)
```

admin_home.html

```
{% include 'admin_header.html'%}


<!-- banner -->
<div class="banner" id="home">
    <div class="banner-overlay-agileinfo">
```

```
<div class="top-header-agile">
<div class="w3l_banner_info">
<section class="slider">
<div class="flexslider">
<ul class="slides">
<li>
<div class="wthree_banner_info_grid">
<h3><span>Genius</span>Welcome to <br>scholar vision</h3>
<p>Your child can be a genius</p>
</div>
</li>
<li>
<div class="wthree_banner_info_grid">
<h3><span>Genius</span>Education <br>for everyone</h3>
<p>Your child can be a genius</p>
</div>
</li>
<li>
<div class="wthree_banner_info_grid">
<h3><span>Genius</span>Welcome to <br>scholar vision</h3>
<p>Your child can be a genius</p>
```

```
            </div>
          </li>
          <li>
            <div class="wthree_banner_info_grid">
              <h3><span>Genius</span>Education <br>for everyone</h3>
              <p>Your child can be a genius</p>
            </div>
          </li>
        </ul>
      </div>
    </section>
  </div>
                </div>
      </div>
</div>


{% include 'footer.html'%}
```

College_home.html

```
{% include 'college_header.html' %}
```

```
<!-- banner -->
<div class="banner" id="home">
    <div class="banner-overlay-agileinfo">
      <div class="top-header-agile">
    <div class="w3l_banner_info">
     <section class="slider">
      <div class="flexslider">
       <ul class="slides">
         <li>
          <div class="wthree_banner_info_grid">
           <h3><span>Genius</span>Welcome to <br>scholar vision</h3>
           <p>Your child can be a genius</p>
          </div>
         </li>
         <li>
          <div class="wthree_banner_info_grid">
           <h3><span>Genius</span>Education <br>for everyone</h3>
           <p>Your child can be a genius</p>
          </div>
```

```
            </li>
            <li>
              <div class="wthree_banner_info_grid">
                <h3><span>Genius</span>Welcome to <br>scholar vision</h3>
                <p>Your child can be a genius</p>
              </div>
            </li>
            <li>
              <div class="wthree_banner_info_grid">
                <h3><span>Genius</span>Education <br>for everyone</h3>
                <p>Your child can be a genius</p>
              </div>
            </li>
          </ul>
        </div>
      </section>
    </div>
            </div>
        </div>
</div>
```

```
{% include 'footer.html'%}


{% include 'admin_header.html' %}


<style>

   th,td{

padding: 20px;

   }

</style>

<center>

   <form method="post">

   <table>

      <tr>

         <th>Reply</th>
         <td><textarea name="reply" id="" cols="60" rows="8" placeholder="Enter Your Reply"
required></textarea></td>

      </tr>

      <td colspan="2" align="center" ><input type="submit" name="replied" ></td>

   </table>

   </form>

</center>
```

```
{% include 'footer.html'%}

<!--footer-->

	<div class="footer w3layouts">

		<div class="container">

			<div class="footer-row w3layouts-agile">

				<div class="col-md-4 footer-grids w3l-agileits">

					<h6><a href="index.html">Scholar Vision</a></h6>

					<p class="footer-one-w3ls">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar tellus sed mauvehicula tempor. </p>

					<div class="top-header-agile-right">

						<ul>

							<li><a href="#"><i class="fa fa-twitter" aria-hidden="true"></i></a></li>

							<li><a href="#"><i class="fa fa-facebook" aria-hidden="true"></i></a></li>

							<li><a href="#"><i class="fa fa-linkedin" aria-hidden="true"></i></a></li>

							<li><a href="#"><i class="fa fa-dribbble" aria-hidden="true"></i></a></li>

						</ul>

					</div>

				<div class="clearfix"></div>

			</div>

			<div class="col-md-2 footer-grids w3l-agileits">
```

```
                    <h3>Footer Menu</h3>

                    <ul class="b-nav">
                        <li><a href="#home">Home</a></li>
                        <li><a href="about.html" >About us</a></li>
                        <li><a href="blog.html">Blog</a></li>
                        <li><a href="contact.html">Contact</a></li>
                    </ul>
                </div>
                <div class="col-md-3 footer-grids w3l-agileits">
                    <h3>Contact Info</h3>
                    <p>Virginia, USA</p>
                    <p>+0 097 338 004</p>
                    <p>El Montee RV, Sterling USA</p>
                    <p><a
href="mailto:info@example.com">mail@example.com</a></p>
                </div>
                <div class="col-md-3 footer-grids w3l-agileits">
                    <h3>Newsletter</h3>
                    <p>It was popularised in the 1960s with the release Ipsum. <p>
                    <form action="#" method="post">
```

```
                                        <input type="email" class="text" required="" />

                                         <input type="submit" value="Go" />


                                    </form>

                                </div>
                                    <div class="clearfix"> </div>

                            </div>

                        </div>

                </div>

<!--//footer-->

<!-- copy-right -->

                        <div class="copyright-wthree">

   <p>&copy; 2017 Scholar Vision . All Rights Reserved | Design by <a
href="http://w3layouts.com/"> W3layouts </a></p>

                            </div>

<!-- //copy-right -->


        <a href="#home" class="scroll" id="toTop" style="display: block;"> <span
id="toTopHover" style="opacity: 1;"> </span></a>

<!-- //smooth scrolling -->



<script type="text/javascript" src="/static/js/jquery-2.1.4.min.js"></script>
```

```
<!-- flexSlider -->

                                                    <script defer
src="/static/js/jquery.flexslider.js"></script>

                                                  <script type="text/javascript">

                                                      $(window).load(function(){

                                                     $('.flexslider').flexslider({

                                                            animation: "slide",

                                                    start: function(slider){

                                                           $('body').removeClass('loading');

                                                             }

                                                      });
                                                    });

                                                </script>

<!-- //flexSlider -->

<!-- requried-jsfiles-for owl -->

 <script src="/static/js/owl.carousel.js"></script>

                                                        <script>

                                                                $(document).ready(function()
                                                                {

                                                                    $("#owl-
demo2").owlCarousel({

                                                                      items : 1,
```

```
                                        lazyLoad : false,
                                        autoPlay : true,
                                        navigation : false,
                                        navigationText :  false,
                                        pagination : true,
                                       });
                                      });
                                     </script>
                                <!-- //requried-jsfiles-for owl -->
<!-- Countdown-Timer-JavaScript -->
                    <script src="/static/js/simplyCountdown.js"></script>
                        <script>
                            var d = new Date(new Date().getTime() + 948 * 120 * 120 * 2000);

                            // default example
                           simplyCountdown('.simply-countdown-one', {
                                    year: d.getFullYear(),
                                month: d.getMonth() + 1,
day: d.getDate()
                                });

                            // inline example
```

```
                simplyCountdown('.simply-countdown-inline', {
                        year: d.getFullYear(),
                    month: d.getMonth() + 1,
                        day: d.getDate(),
                inline: true
                        });

                    //jQuery example
                $('#simply-countdown-losange').simplyCountdown({
                        year: d.getFullYear(),
                    month: d.getMonth() + 1,
                day: d.getDate(),
        enableUtc: false
                        });
                </script>
            <!-- //Countdown-Timer-JavaScript -->


<!--search-bar-->
            <script src="/static/js/main.js"></script>
<!--//search-bar-->
```

```
 <!-- start-smoth-scrolling -->
<script type="text/javascript" src="/static/js/move-top.js"></script>

<script type="text/javascript" src="/static/js/easing.js"></script>

<script type="text/javascript">

	jQuery(document).ready(function($) {

			$(".scroll").click(function(event){

event.preventDefault();

			  $('html,body').animate({scrollTop:$(this.hash).offset().top},1000);

			});

		});

</script>

<!-- start-smoth-scrolling -->

<!-- here stars scrolling icon -->

	<script type="text/javascript">

			$(document).ready(function() {

				/*

					var defaults = {

					  containerID: 'toTop', // fading element id

				containerHoverID: 'toTopHover', // fading element hover id
```

```
    scrollSpeed: 1200,     easingType: 'linear'

                                    };

                                */


                            $().UItoTop({ easingType: 'easeOutQuart' });


                                });

        </script>

<!-- //here ends scrolling icon -->
<!--js for bootstrap working-->

     <script src="/static/js/bootstrap.js"></script>

<!-- //for bootstrap working -->

</body>

</html>
```

SYSTEM TESTING

SYSTEM TESTING

Testing is the process that should be done during the development process. It offers the greatest security since the old system can take over the errors is found or inability to handle certain type of transactions System testing is the evaluation of software item to detect difference between given input and expected output. Also to assess the feature of software item Testing assesses the quality of the product. Software while using the new system. So testing is vital to the success of new system. Series of the system are performed for the proposed system before the system is ready for user acceptance. In other word software testing is a verification and validation process.

6.1 TESTING METHODOLOGIES

These exists different methodologies in testing to make sure phase .And to check

whether it satisfies the specified requirements at the end of the development phase.

The major testing strategies are

• Unit Testing

Integrated Testing

System Testing

•Acceptance Testing

6.1.1 Unit Testing

•Unit test is also known as code functional testing is a structural testing which is actually run by the computer on the built product.

Unit test is initial part of structural testing corresponding to some quick checks that developer perform before subjecting the code more extensive code coverage testing.

• Unit test can run the program under a debugger or an integrated development environment.

• Unit test which is box oriented and the step can be concluded in parallel for component.

6.1.2. Integrated testing

Integration is defined as the set of interaction among components. Testing the interaction between the interactions with the other system externally is called integrated testing

• Integration testing focuses on testing interface that are "implicit and explicit" and "internal and external".

• Internal interface provide communication across two modules with in a project and external interface are visible outside the product of the third party developers and solution providers.
• Integration is done with test cases, which test the functionality of the software.

Integration testing both looks for both implicit and explicit interfaces and tests all these interaction.

Integration testing approach could be termed as the gray test boxing approach.

In our system there exists a set of interaction between modules.

We test these interaction through valid ids (primary keys). So we can ensure that there exists a successful interaction between modules.

6.1.3. System Testing

•System testing is performed on the basis of written test cases according to information collected from detailed architecture/design document, module specification, and requirement specification.

•System testing may be started one unit, component, and integration testing are completed.

In our project we test the complete system by evaluating the system compliance with the specified requirement. So we can ensure that the meets the all the requirement mentioned by the customer.

IMPLEMENTATION

IMPLEMENTATION

7.1 INTRODUCTION TO SYSTEM IMPLEMENTATION

Once the system has checked and performs its operations successfully, it can be put into operation. It involves a computer compatible file, training the operating staff, installing hardware, terminals and telecommunication network etc. Implementation is the stage of the project where the theoretical design is turn in to a working system. The implementation stage is a system project. It involves careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the change over, and the evolution method. Once the planning has been completed, the major effort is to ensure that the programs in the system are working properly. At the same time concentrate on training user staff. The implementation phase is an important one in which the source code put in to the operation. Before implementing the software, careful testing and documentation is necessary. During the implementation and testing phases the configuration management and quality assurance of requirements, design specification and source code are performed. Implementation should provide with well-defined software requirements, design specifications. The major milestone for project implementation is successful integration of software components in the functioning system.

The term implementation has different meanings, ranging from the conversion of a basic application to a complete replacement of a system. The procedure, however, is virtually the same. Implementation is used here to mean the process of converting a new or revised system design to an operational one. 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.

7.2 SYSTEM MAINTENANCE

Once the software is fully developed and implemented, the company starts to use the software. The company also grows, and more divisions can be attached to the company, or the database of

the company can grow. So, after some time the software, which has been installed, needs some modification. If the software needs modification all the steps needed to develop new software must be executed. Maintenance can be classified as corrective, adaptive or perceptive.

[Corrective maintenance: Reactive modification of a software product performed after delivery to correct discovered problems.

CAdaptive maintenance: Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.

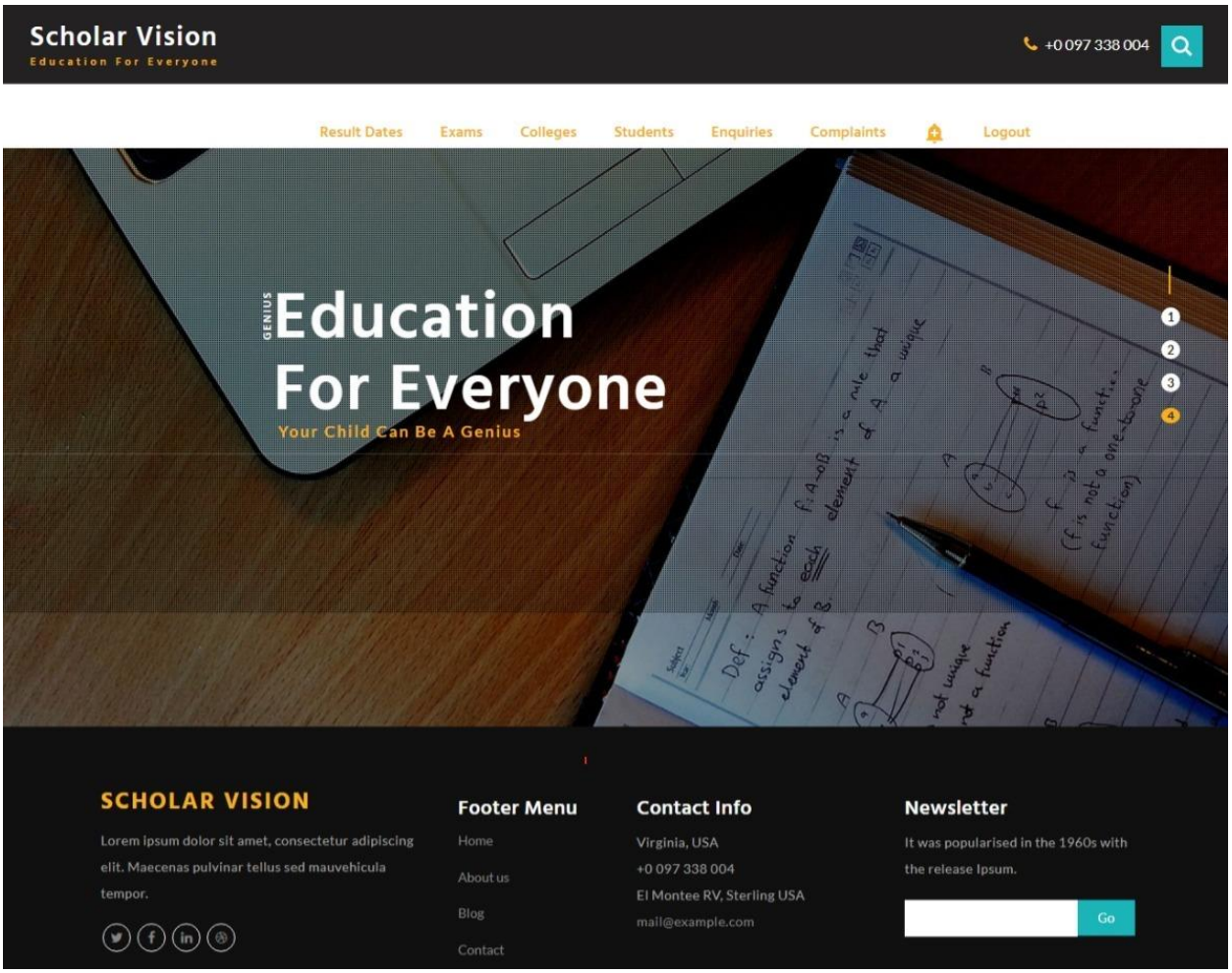
Perfective maintenance: Modification of a software product after delivery to improve performance or maintainability.

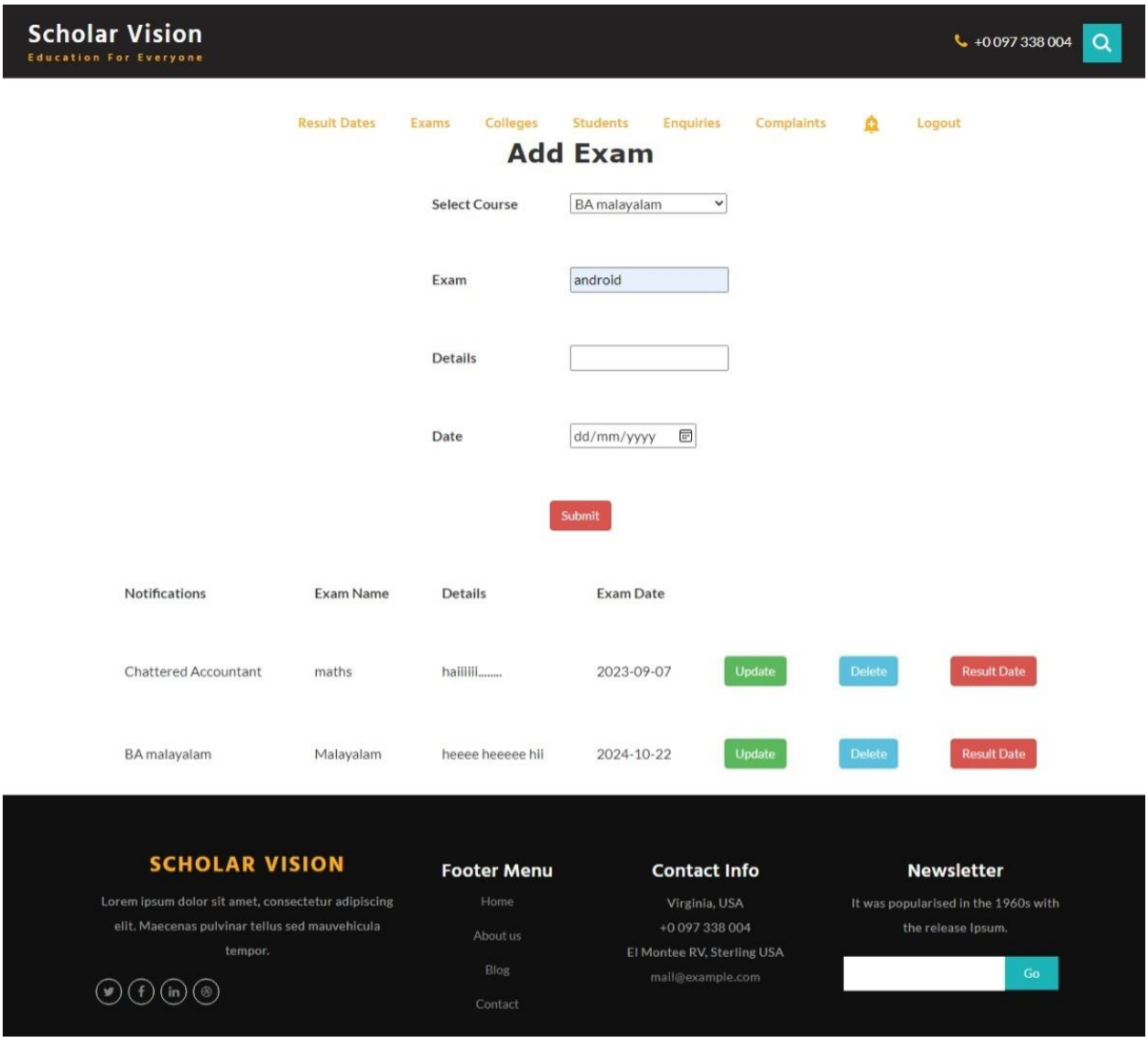
8. CONCLUSION

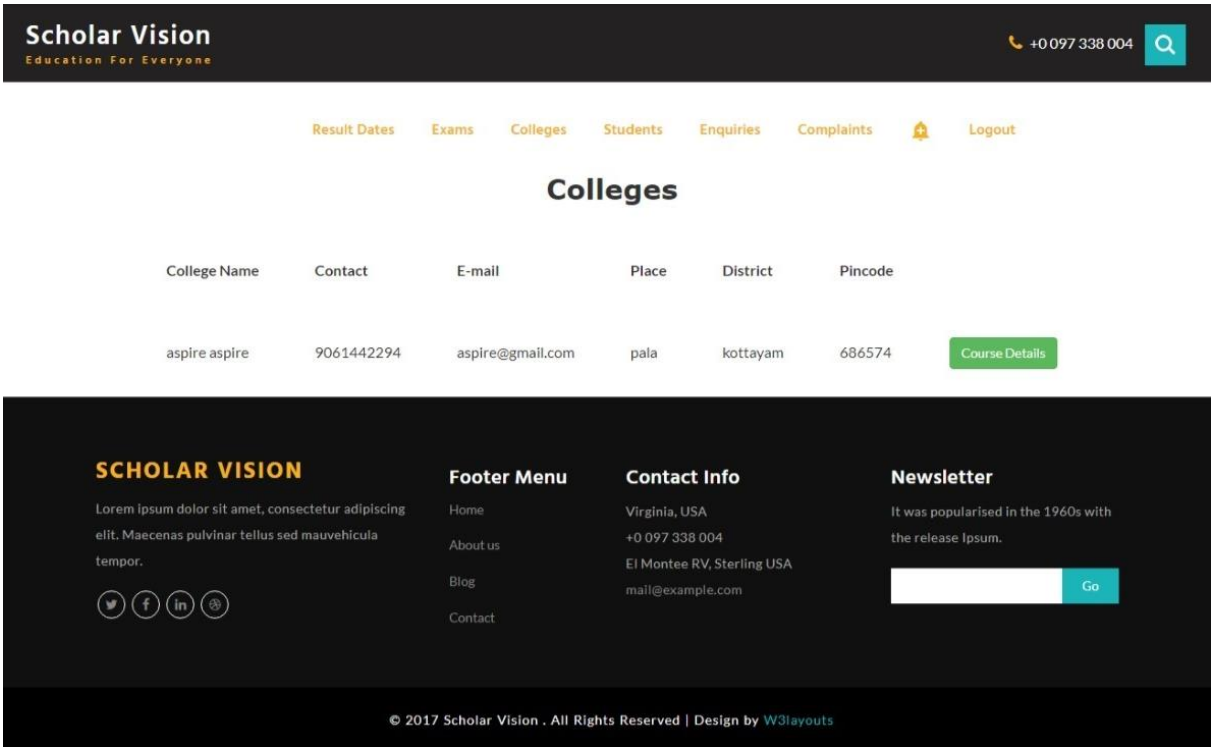
CONCLUSION

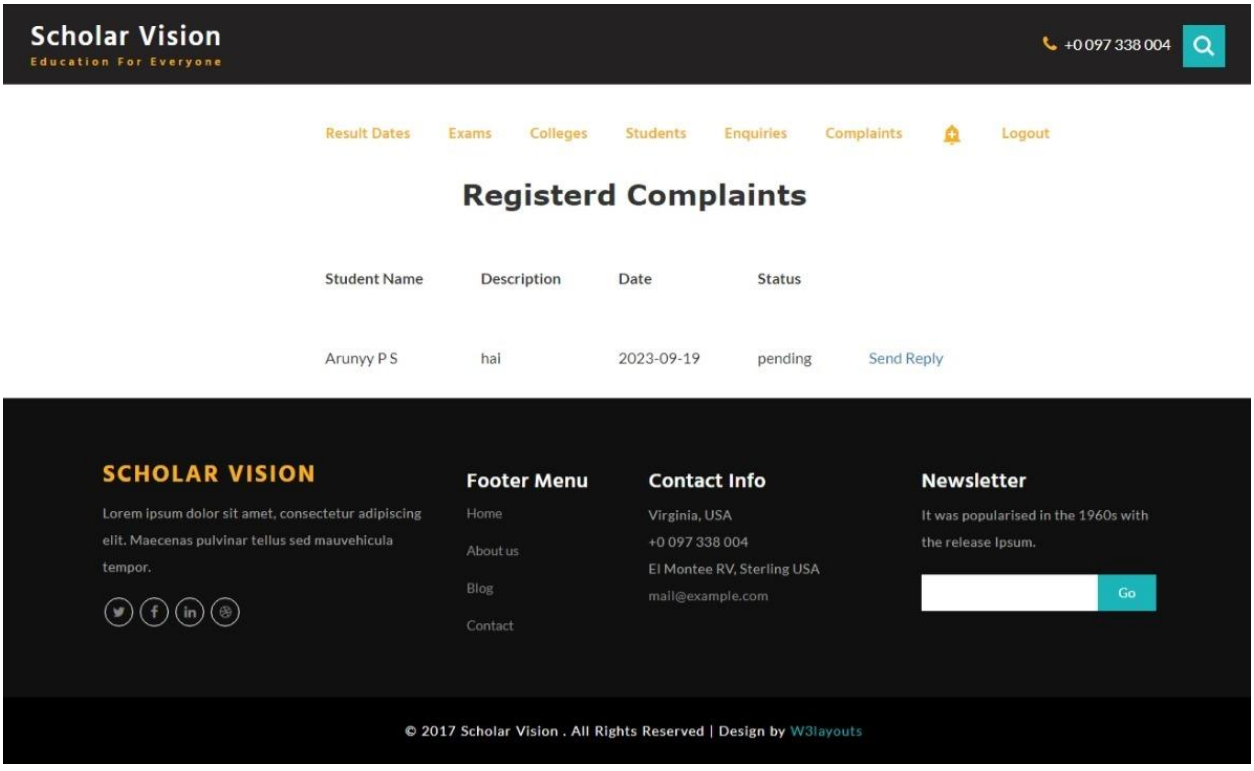
CollegeConnect revolutionizes the college experience by offering an integrated mobile platform that connects students, faculty, and staff seamlessly. By promoting collaboration, communication, and easy access to essential resources, the app fosters a vibrant college community. CollegeConnect's dynamic features empower students to take charge of their academic journey, engage in extracurricular activities, and stay informed about campus happenings. With its user-centric approach, CollegeConnect is set to become an indispensable tool for enhancing the overall college life and strengthening the bond within the college ecosystem.

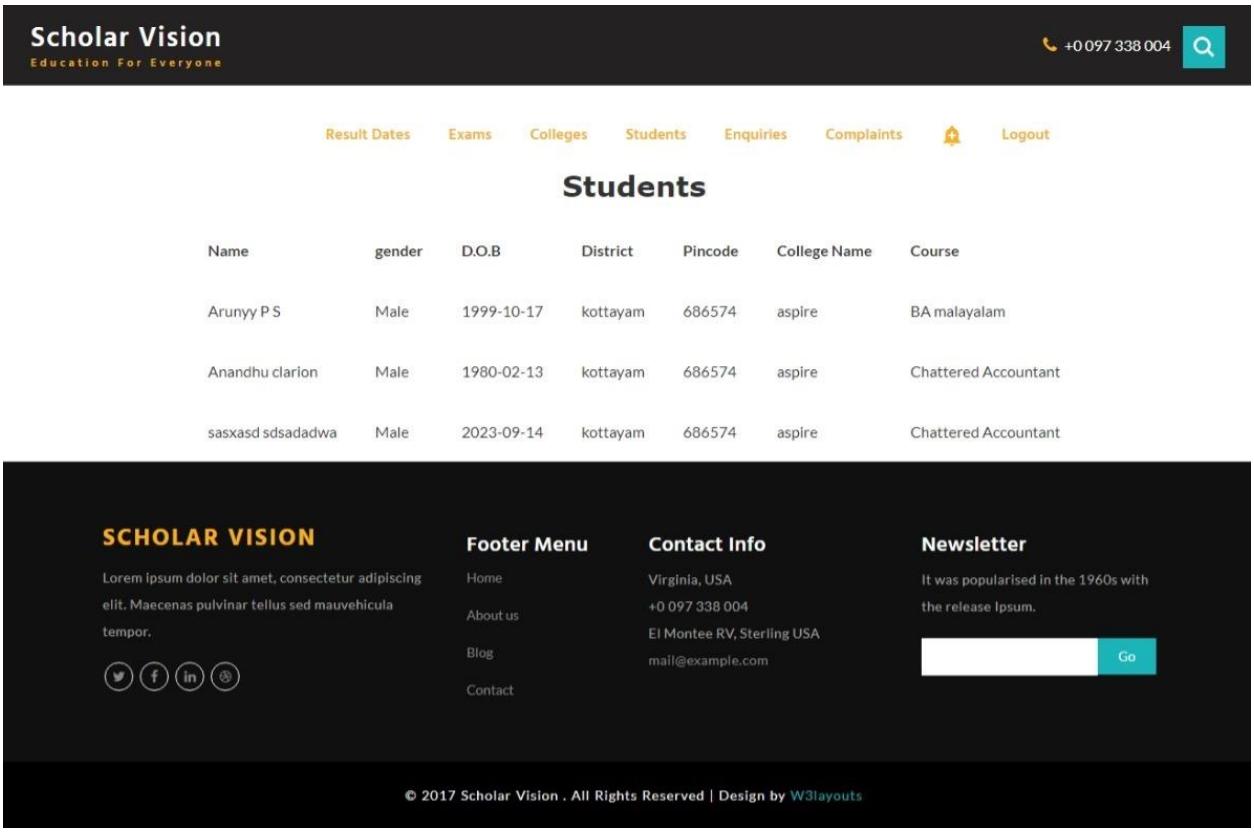
APPENDIX

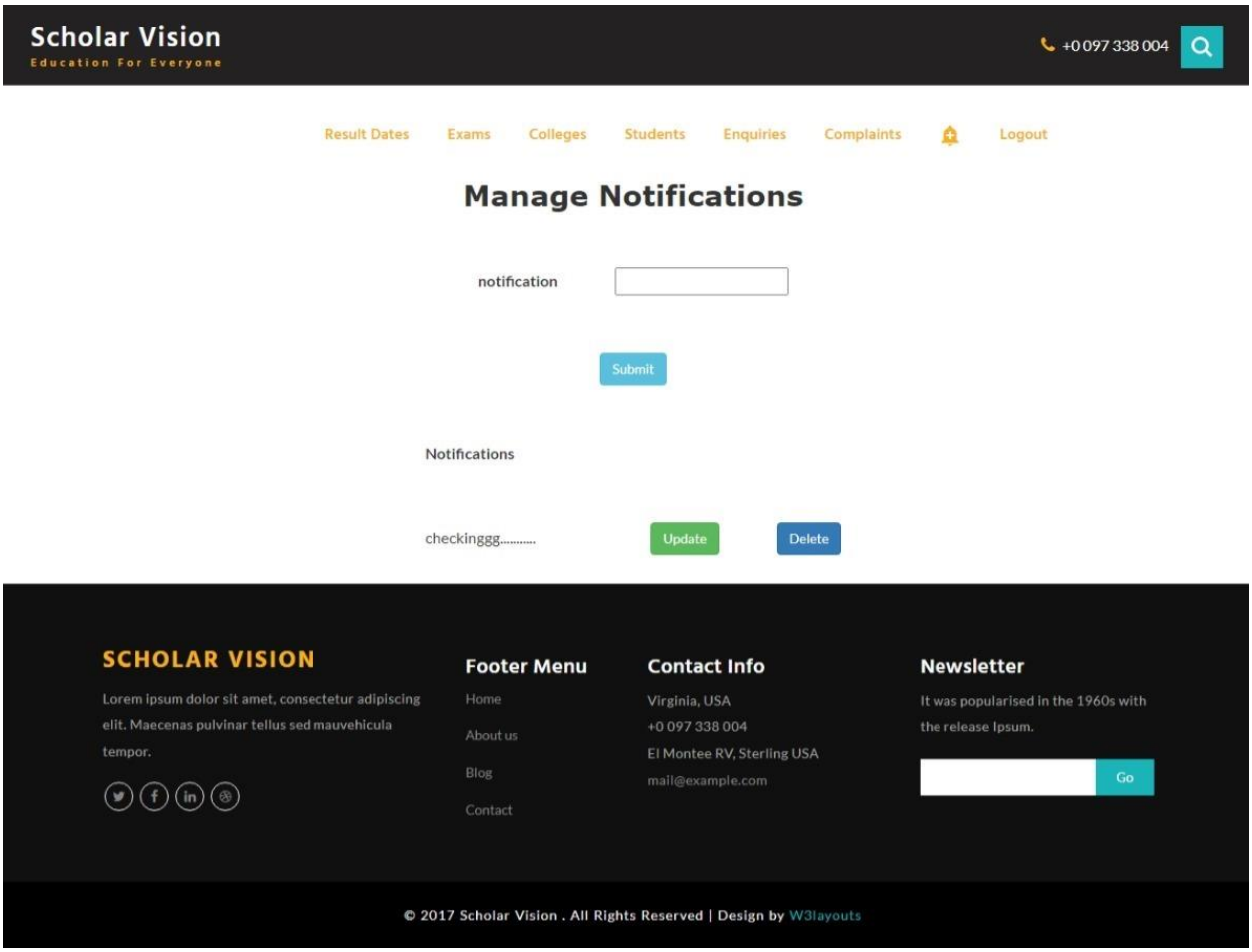

Scholar Vision
Education For Everyone
+0 097 338 004
Result Dates
Exams
Colleges
Students
Enquiries
Complaints
Logout
GENIUS
Education
For Everyone
Your Child Can Be A Genius
1
2
3
4
SCHOLAR VISION
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar tellus sed mauvehicula tempor.
Footer Menu
Home
About us
Blog
Contact
Contact Info
Virginia, USA
+0 097 338 004
El Montee RV, Sterling USA
mail@example.com
Newsletter
It was popularised in the 1960s with the release Ipsum.
Go

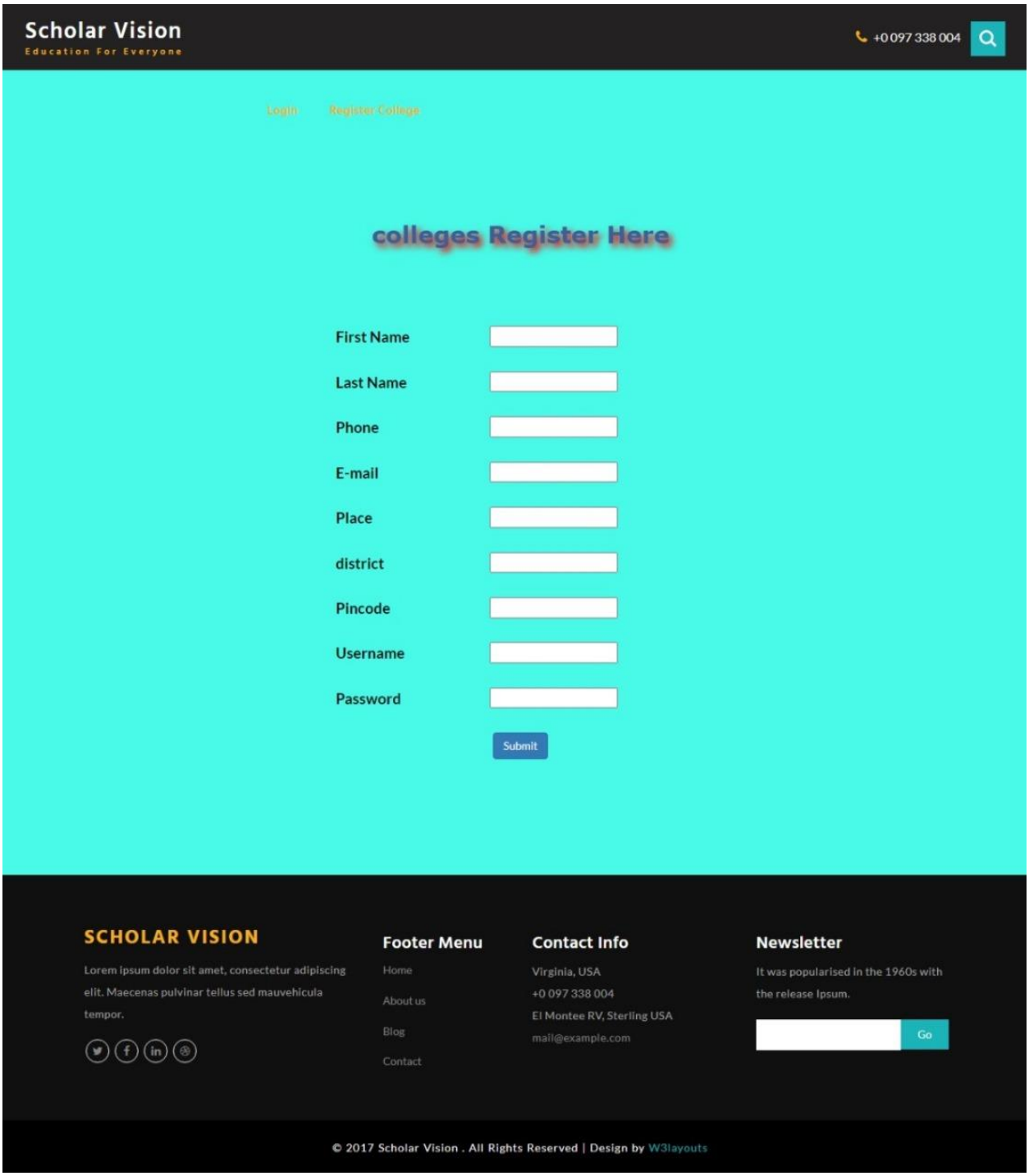

Scholar Vision
Education For Everyone
+0 097 338 004
Result Dates
Exams
Colleges
Students
Enquiries
Complaints
Logout
Add Exam
Select Course
BA malayalam
Exam
android
Details
Date
dd/mm/yyyy
Submit
Notifications
Exam Name
Details
Exam Date
Chattered Accountant
maths
haiiiiii.....
2023-09-07
Update
Delete
Result Date
BA malayalam
Malayalam
heeee heeeee hii
2024-10-22
Update
Delete
Result Date
SCHOLAR VISION
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar tellus sed mauvehicula tempor.
Footer Menu
Home
About us
Blog
Contact
Contact Info
Virginia, USA
+0 097 338 004
El Montee RV, Sterling USA
mail@example.com
Newsletter
It was popularised in the 1960s with the release Ipsum.
Go


Scholar Vision
Education For Everyone
+0 097 338 004
Result Dates
Exams
Colleges
Students
Enquiries
Complaints
Logout
Colleges
College Name
Contact
E-mail
Place
District
Pincode
aspire aspire
9061442294
aspire@gmail.com
pala
kottayam
686574
Course Details
SCHOLAR VISION
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar tellus sed mauvehicula tempor.
Footer Menu
Home
About us
Blog
Contact
Contact Info
Virginia, USA
+0 097 338 004
El Montee RV, Sterling USA
mail@example.com
Newsletter
It was popularised in the 1960s with the release Ipsum.
Go
© 2017 Scholar Vision . All Rights Reserved | Design by W3layouts


Scholar Vision
Education For Everyone
+0 097 338 004
Result Dates
Exams
Colleges
Students
Enquiries
Complaints
Logout
Registerd Complaints
Student Name
Description
Date
Status
Arunyy P S
hai
2023-09-19
pending
Send Reply
SCHOLAR VISION
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar tellus sed mauvehicula tempor.
Footer Menu
Home
About us
Blog
Contact
Contact Info
Virginia, USA
+0 097 338 004
El Montee RV, Sterling USA
mail@example.com
Newsletter
It was popularised in the 1960s with the release Ipsum.
Go
© 2017 Scholar Vision . All Rights Reserved | Design by W3layouts


Scholar Vision
Education For Everyone
+0 097 338 004
Result Dates
Exams
Colleges
Students
Enquiries
Complaints
Logout
Students
Name gender D.O.B District Pincode College Name Course
Arunyy P S Male 1999-10-17 kottayam 686574 aspire BA malayalam
Anandhu clarion Male 1980-02-13 kottayam 686574 aspire Chattered Accountant
sasxasd sdsadadwa Male 2023-09-14 kottayam 686574 aspire Chattered Accountant
SCHOLAR VISION
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar tellus sed mauvehicula tempor.
Footer Menu
Home
About us
Blog
Contact
Contact Info
Virginia, USA
+0 097 338 004
El Montee RV, Sterling USA
mail@example.com
Newsletter
It was popularised in the 1960s with the release Ipsum.
Go
© 2017 Scholar Vision . All Rights Reserved | Design by W3layouts


Scholar Vision
Education For Everyone
+0 097 338 004
Result Dates
Exams
Colleges
Students
Enquiries
Complaints
Logout
Manage Notifications
notification
Submit
Notifications
checkinggg.....
Update
Delete
SCHOLAR VISION
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar tellus sed mauvehicula tempor.
Footer Menu
Home
About us
Blog
Contact
Contact Info
Virginia, USA
+0 097 338 004
El Montee RV, Sterling USA
mail@example.com
Newsletter
It was popularised in the 1960s with the release Ipsum.
Go
© 2017 Scholar Vision . All Rights Reserved | Design by W3layouts


Scholar Vision
Education For Everyone
+0 097 338 004
Login
Register College
colleges Register Here
First Name
Last Name
Phone
E-mail
Place
district
Pincode
Username
Password
Submit
SCHOLAR VISION
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar tellus sed mauvehicula tempor.
Footer Menu
Home
About us
Blog
Contact
Contact Info
Virginia, USA
+0 097 338 004
El Montee RV, Sterling USA
mail@example.com
Newsletter
It was popularised in the 1960s with the release Ipsum.
Go
© 2017 Scholar Vision . All Rights Reserved | Design by W3layouts

BIBLOGRAPHY

Websites

• www.wschool.com

www.wikipedia.com www.immunize.org

www.javatpoint.com www.final-

yearproject.com

• www.grapestech.com/vaccination www.tutorialspoint.com

• www.softwaretestinghelp.com

Books

> Fundamentals of Software Engineering fourth edition by Rajib Mall

> Head First Android Development second edition by Dawn & David

• Professional Android fourth edition by Reto Meier & Ian Lake

> Android Studio Development fifth edition by Neil Smyth

• Software Testing by Gopalaswamy Ramesh & Stinivasan Desikan