VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA” BELAGAVI 590014

A DBMS Mini–Project Report

on

“Computer Classes Registration System”

BACHELOR OF ENGINEERING
In
Department of Computer Science and Engineering

Submitted By

RAHUL D R [4GM21CS075]

As a part of curriculum for

DBMS Laboratory with Mini Project - Subject code: ***21CSL55***

Submitted to

Dr. Maheswari L Patil
Faculty In-charge

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GM INSTITUTE OF TECHNOLOGY, DAVANGERE

(Affiliated to VTU, Belagavi, Approved by AICTE -New Delhi & Govt. of Karnataka)

(Accredited by NBA New Delhi, Valid up to 30.06.2025)

2023-24

Srishyla Educational Trust (R), Bheemasamudra

GM INSTITUTE OF TECHNOLOGY, DAVANGERE

(Affiliated to VTU, Belagavi, Approved by AICTE -New Delhi & Govt. of Karnataka)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Accredited by NBA New Delhi, Valid up to 30.06.2025)

CERTIFICATE

This is to certify that the DBMS Mini Project work entitled **"COMPUTER CLASSES REGISTRATION SYSTEM"** carried out by **RAHUL D R (4GM21CS075)** Bonafede student of GMIT, Davangere. The Mini Project work carried out as a part of curriculum for 5th Semester course Database Management Laboratory with Mini Project having subject code 21CSL58, in department of Computer Science and Engineering, as per VTU, Belagavi for the academic year 2023-2024. It is certified that all the corrections and suggestions indicated for Internal assessment have been incorporated in the Report. The report prepared and project work carried out is satisfactory

---------------------------------- ----------------------------------

Dr. Maheswari L Patil — *Faculty In-charge*

Dr. B N Veerappa — *Head of the Department*

External Viva

Name of the Examiners **Signature with Date**

1.

2

ABSTRACT

The Computer Classes Registration System (EnrollEssy) is a web-based platform designed to streamline the process of registering for computer classes offered by educational institutions. Built using Python for the backend and Django as the web framework, the system provides an intuitive interface for students to browse available courses, enrol in desired classes, and manage their registrations.

Key features of the system include course categorization, instructor management, enrolment tracking, and secure user authentication. Administrators can efficiently manage course offerings, monitor enrolment status, and generate insightful reports to aid in decision-making. The system prioritizes data integrity, security, and scalability, ensuring the confidentiality of student information and the reliability of course management operations.

By leveraging Python's versatility and Django's robustness, the Computer Classes Registration System aims to enhance the user experience, optimize administrative processes, and foster a conducive learning environment for students pursuing computer education.

LIST OF FIGURES

CONTENTS

CHAPTER-1

INTRODUCTION

1.1 Database management system

A database management system (DBMS) refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient.

Database systems are meant to handle large collections of information. Management of data involves both defining structures for the storage of information and providing mechanisms that can do the manipulation those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

1.2 Problem Statement

The Computer Classes Registration System aims to streamline the process of managing registrations, course enrollments, and student information for computer classes offered by an educational institution or training center. The system will provide an efficient platform for both administrators and students to facilitate the registration process and manage course enrollments effectively.

1.3 Objective

The objective of the Computer Classes Registration System is to create a user-friendly and efficient platform that simplifies the process of registering for computer classes, improves course management, and enhances the overall student experience. By automating manual tasks and providing real-time access to course information, the system aims to streamline operations, reduce administrative overhead, and improve the quality of service for both students and administrators.

CHAPTER 2

DATABASE DESIGN

2.1 System Requirements:

2.1.1 Software Requirement:

The software requirements for the development of this project are:

- Software: Django
- Operating System: Windows 7,9,10,11
- Front End: HTML, CSS, JavaScript
- Programming Language: Python
- Data Base Environment: Object-Relational Mapping (ORM)
- Server: default wsgi server

2.1.2 Hardware Requirements:

The hardware required for the development of this project are:

- **Processor:** Standard processor with 1.6 GHz or more
- **RAM:** 512MB or more
- **Hard Disk:** 20GB or more
- **Monitor:** Standard Color Monitor

CONCEPTUAL DESIGN

2.2 E–R DIAGRAM:

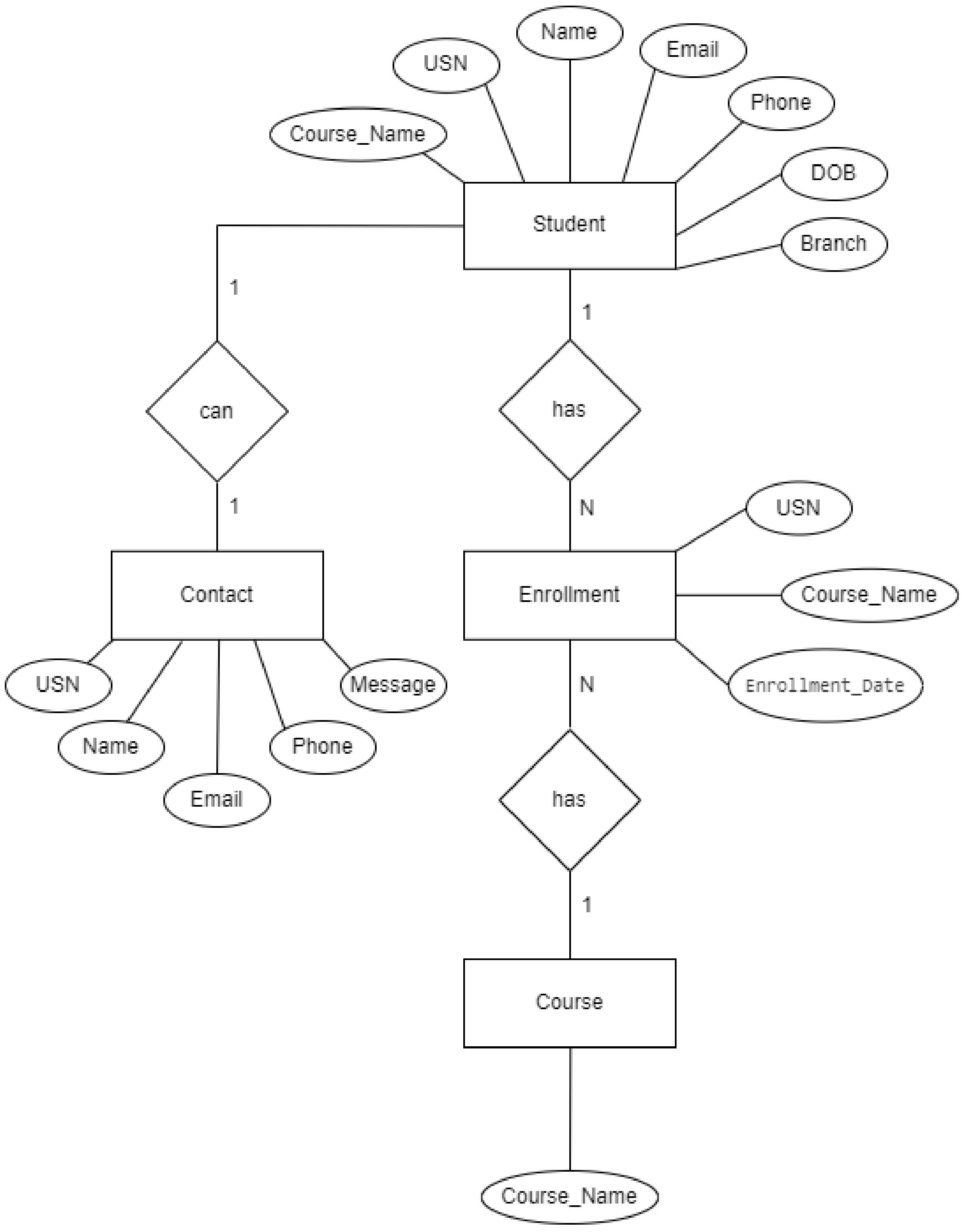


Fig 2.2.1: E-R Diagram of Computer Classes Registration System

2.3 SCHEMA DIAGRAM:

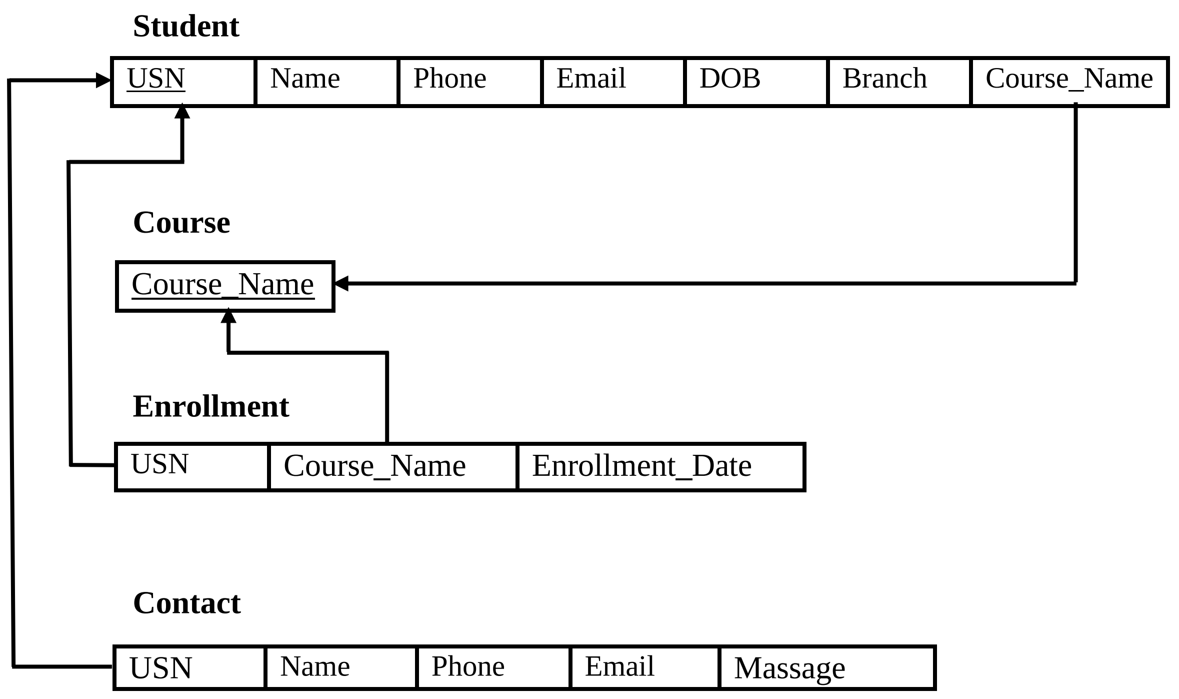


Fig 2.3.1: Schema Diagram of Computer Classes Registration System

CHAPTER 3

IMPLEMENTATION

3.1 FRONT END:

- **HTML**:

  HTML stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. HTML 5 is the fifth and current version of HTML. It has improved the markup available for documents and has introduced application programming interfaces (API) and Document Object Model (DOM)

- **CSS**:

  Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.

- **JAVASCRIPT**:

  JavaScript s a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications.

3.2 BACK END:

For the backend of the Computer Classes Registration System, Python will be used as the primary programming language. Python offers a wealth of frameworks and tools for web development, making it a popular choice for building the server-side logic and managing databases.

Key components of the backend implementation using Python include:

- **Framework:** Django will be utilized as the web framework for the backend. Django provides a powerful and feature-rich environment for building web applications rapidly and efficiently. It includes built-in components for handling URL routing, database interactions, user authentication, and more.

- **Database:** Django supports various relational database management systems (RDBMS) such as PostgreSQL, MySQL, SQLite, and Oracle. The choice of database will depend on factors like scalability, performance, and project requirements.

- **ORM (Object-Relational Mapping):** Django's ORM simplifies the interaction with the database by allowing developers to work with database models using Python classes and objects. This abstraction layer facilitates database operations without writing raw SQL queries.

- **Views and Controllers:** In the Django framework, views represent the logic behind the web pages and controllers handle user requests. Python functions or classes are used to define views, which process requests, interact with models, and return responses to the client.

- **RESTful APIs (Application Programming Interfaces):** If needed, Django can be used to create RESTful APIs for integrating the system with other applications or services. Django Rest Framework (DRF) is a powerful toolkit for building APIs in Django, providing serialization, authentication, and other features out of the box.

- **Middleware:** Django middleware allows developers to modify request and response objects, perform authentication, logging, and other tasks at the middleware level.

- **Security:** Django includes built-in security features such as protection against common web vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

- **Session Management and Authentication:** Django provides session management and authentication mechanisms to handle user sessions, user authentication, permissions, and access control.

3.3 SQL CODE IMPLEMENTATION:

Models.py

```
from django.db import models

# Create your models here.
class Course(models.Model):
    Course_Name = models.CharField(max_length=100, primary_key=True)

    def __str__(self):
        return self.Course_Name


class Student(models.Model):
    Course_Name = models.ForeignKey(Course, on_delete=models.CASCADE)
    USN = models.CharField(max_length=10, primary_key=True)
    name = models.CharField(max_length=20)
    Email = models.CharField(max_length=50)
    Phone = models.CharField(max_length=50,null=True)
    DOB = models.DateField(null=True, blank=True)
    Branch = models.CharField(max_length=20,null=True)

    def __str__(self):
        return f"{self.name} - {self.USN} - {self.Course_Name}"


class Enrollment(models.Model):
    USN = models.ForeignKey(Student, on_delete=models.CASCADE)
    Course_Name = models.ForeignKey(Course, on_delete=models.CASCADE)
    Enrollment_Date = models.DateField()

    def __str__(self):
        return f"{self.USN}"

class Contact(models.Model):
    name = models.CharField(max_length=20)
    email = models.CharField(max_length=50)
    phone = models.CharField(max_length=50,null=True)
    massege = models.CharField(max_length=1000,null=True)

    def __str__(self):
        return self.name
```

Views.py

```
from django.shortcuts import render, redirect, HttpResponse
from datetime import datetime
from Home.models import Contact, Student
from django.contrib import messages

def index(request):
    return render(request, 'index.html')

def course(request):
    return render(request, 'course.html')


def about(request):
    return render(request, 'about.html')

def faq(request):
    return render(request, 'faq.html')

def contact(request):
    if request.method == 'POST':

        name = request.POST.get("name")
        email = request.POST.get("email")
        phone = request.POST.get("phone")
        massege = request.POST.get("massege")

        # Corrected to use Contact.objects.get_or_create()
        contact=Contact.objects.create(
            name=name,
            email=email,
            phone=phone,
            massege=massege
        )
        contact.save()
        messages.success(request, "Your Form has been sent")
    return render(request, 'contact.html')

def student(request):
    if request.method == 'POST':
        Course_Name = request.POST.get("Course_Name")
        USN = request.POST.get("USN")
        name = request.POST.get("name")
        Email = request.POST.get("Email")
        Phone = request.POST.get("Phone")
        DOB = request.POST.get("DOB")
        Branch = request.POST.get("Branch")

        # Corrected to use Contact.objects.get_or_create()
        student=Student.objects.create(
            Course_Name = Course_Name,
            USN=USN,
            name=name,
            Email=Email,
            Phone=Phone,
            DOB=DOB,
            Branch=Branch
        )
        student.save()
        messages.success(request, "Your Form has been sent")
    return render(request, 'student.html')
```

3.4 OUTCOMES

Outcomes of our system:

- **Efficient Registration Process**: The system should streamline the registration process for students interested in computer classes. This means reducing paperwork, minimizing errors, and making it easy for students to sign up for courses online.
- **Improved Communication**: The system can facilitate better communication between students, instructors, and administrators. It should provide timely notifications about course updates, schedules, and important announcements.

CHAPTER 4

ADMIN SITE SNAPSHORT

4.1 LOGIN PAGE:

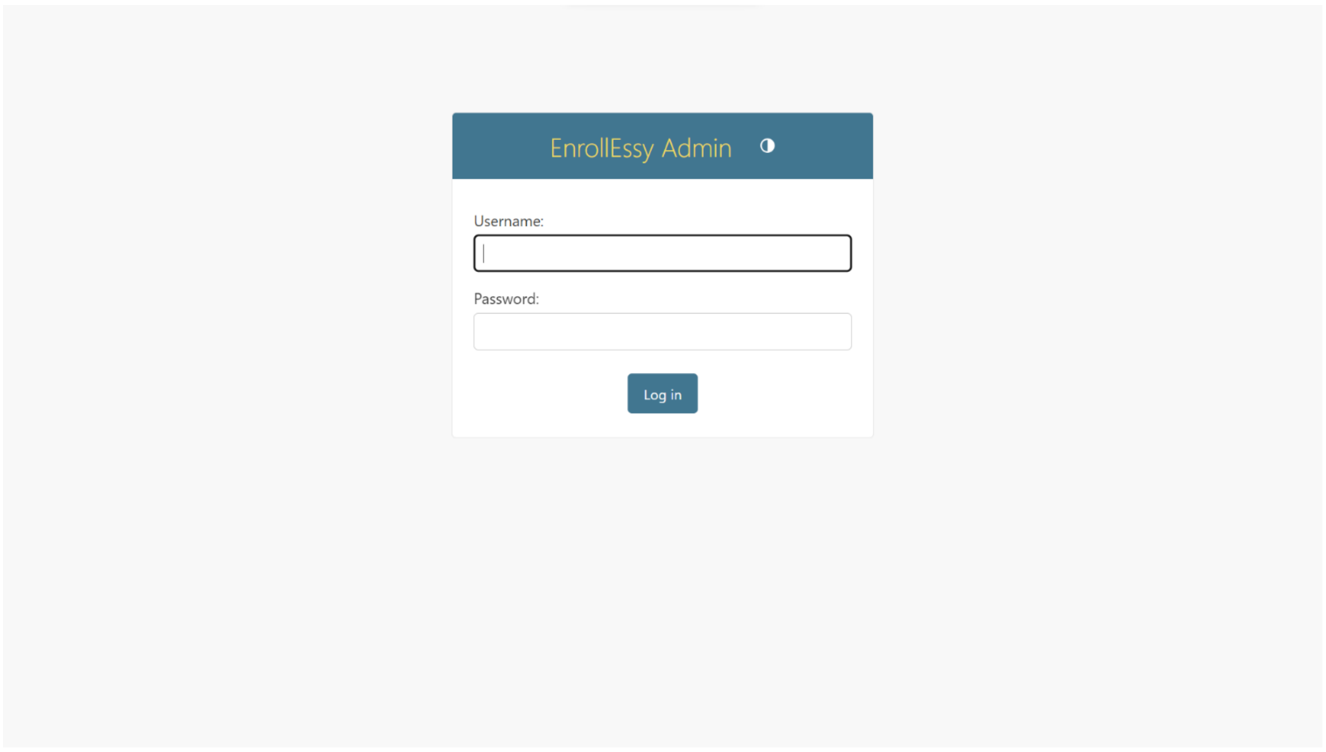
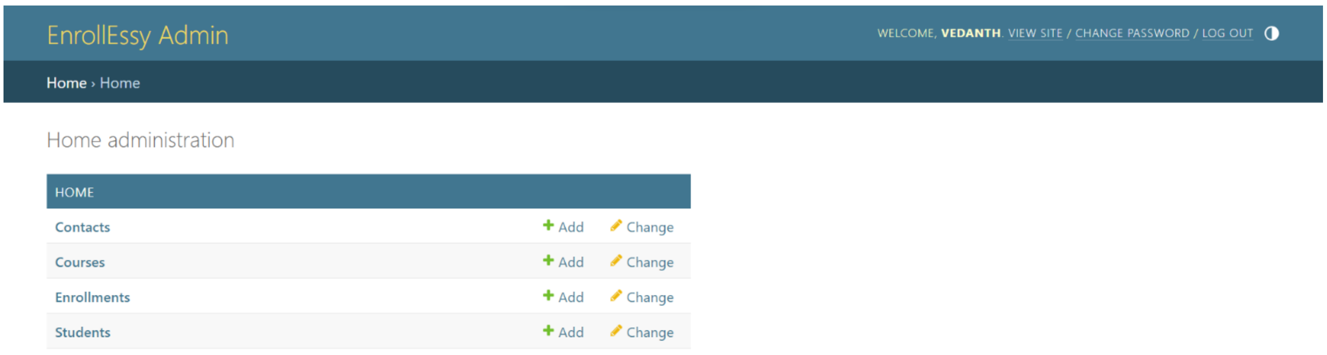


Fig 4.1: Admin Login Page

4.2 HOME PAGE:


EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

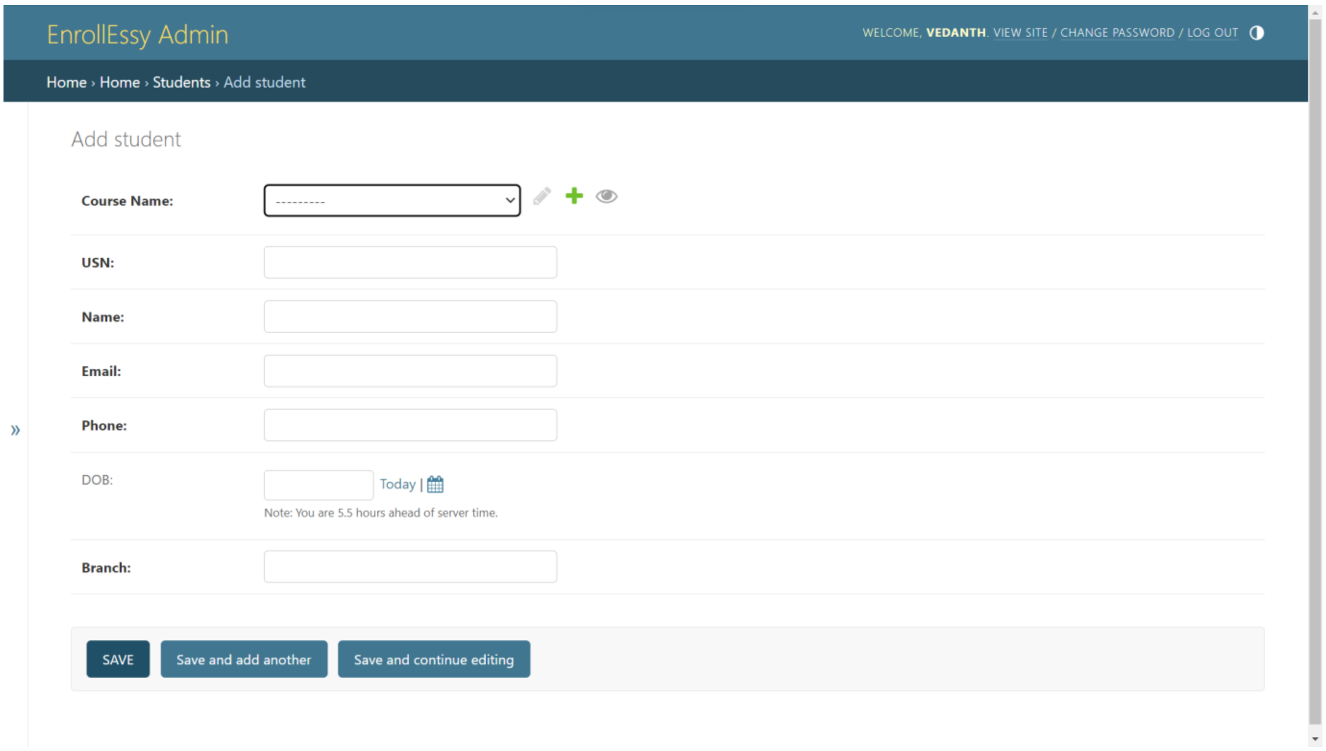
Home › Home

Home administration

HOME		
Contacts	Add	Change
Courses	Add	Change
Enrollments	Add	Change
Students	Add	Change

Fig 4.2: Admin Home Page

4.3 TABLES:
4.3.1 STUDENT:



EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

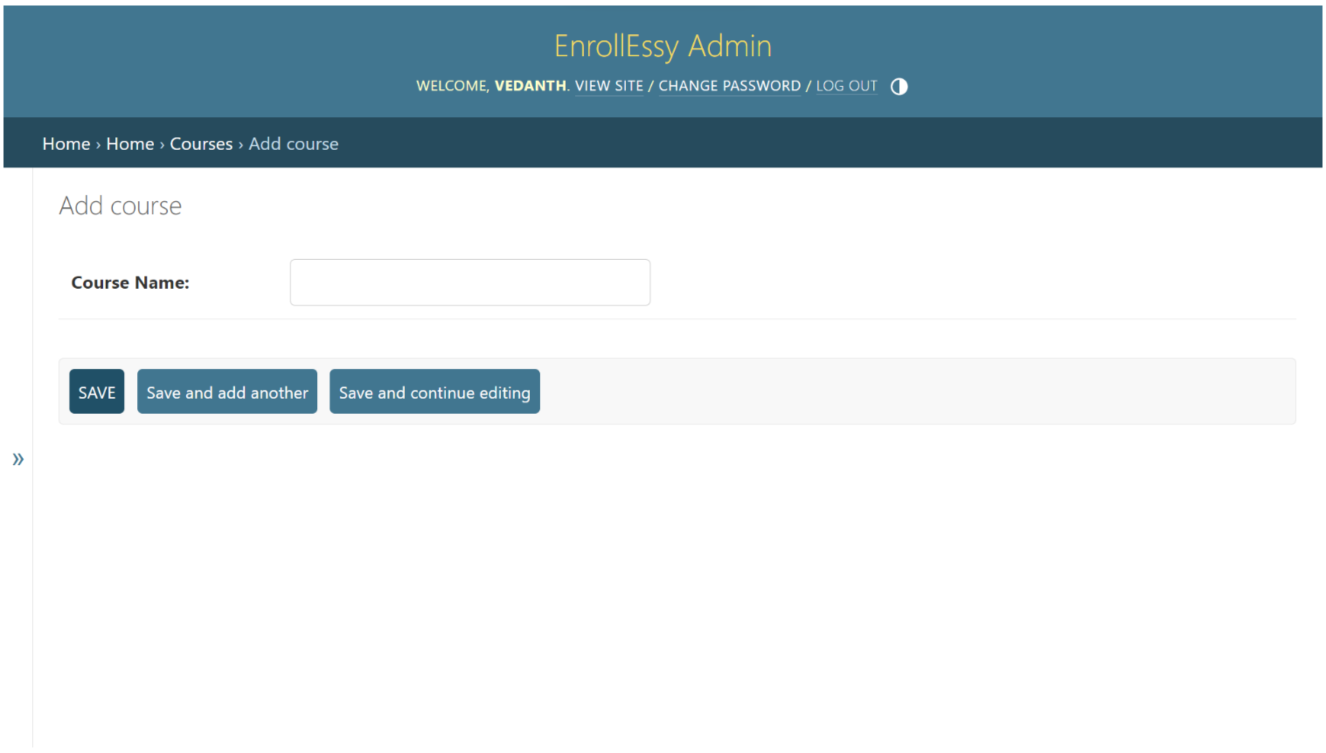
Home › Home › Students › Add student

Add student

Course Name:	---------
USN:	
Name:	
Email:	
Phone:	
DOB:	Today | Note: You are 5.5 hours ahead of server time.
Branch:	

Fig 4.3: Student Table

4.3.2 COURSE:



EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

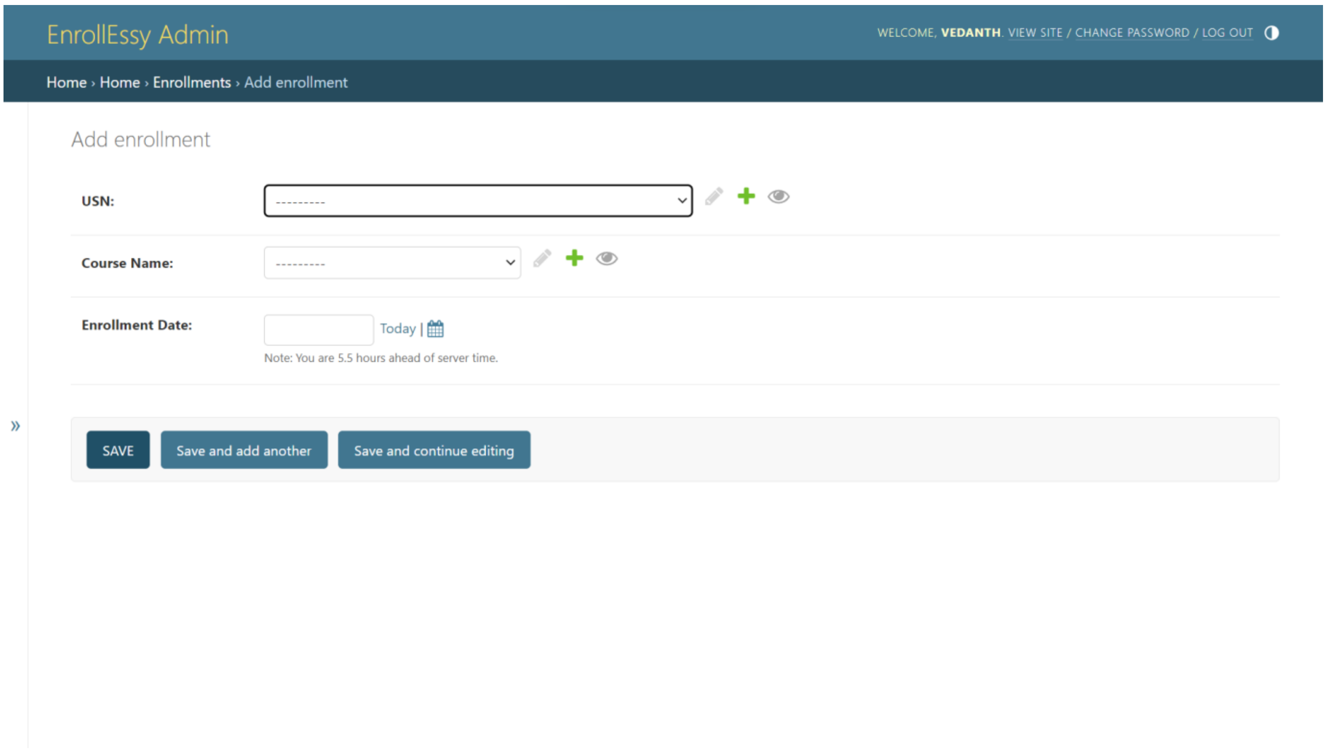
Home › Home › Courses › Add course

Add course

Course Name:

Fig 4.4: Course Table

4.3.3 ENROLLMENT


EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

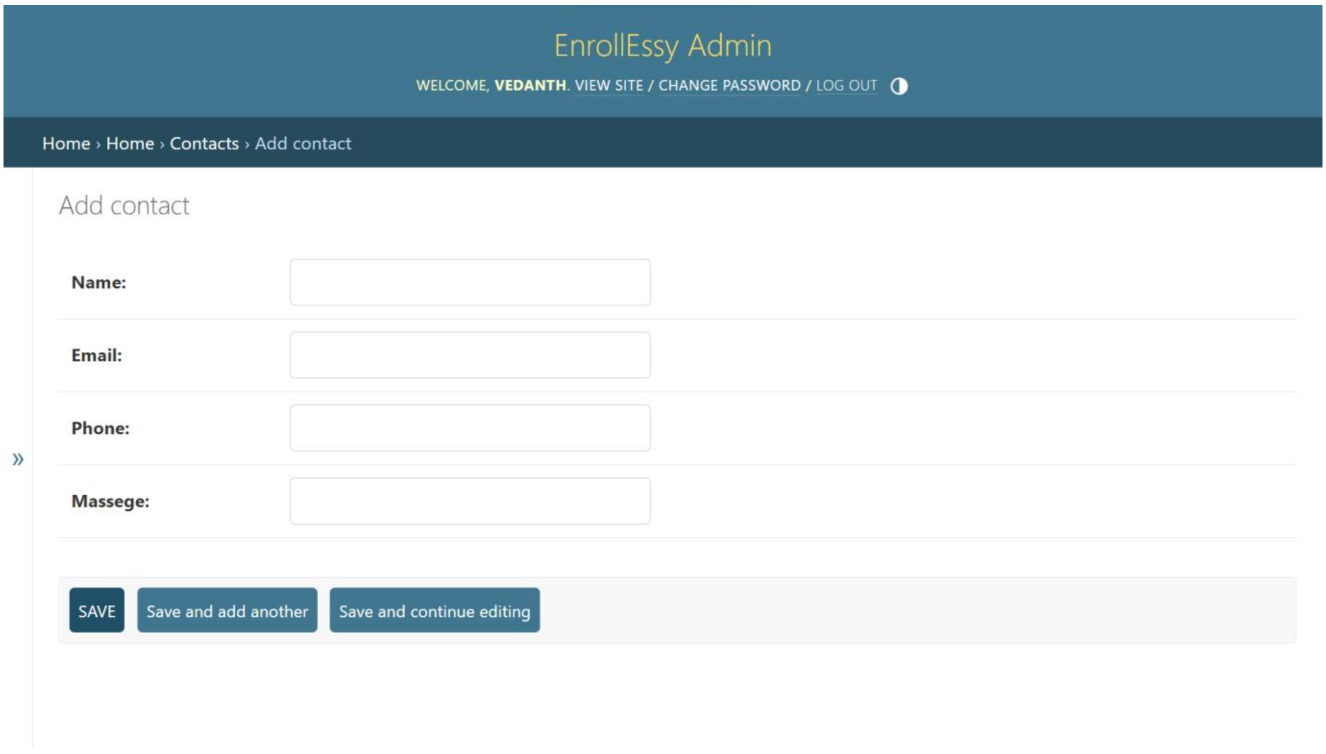
Home › Home › Enrollments › Add enrollment

Add enrollment

USN:	---------
Course Name:	---------
Enrollment Date:	Today | Note: You are 5.5 hours ahead of server time.

Fig 4.5: Enrollment Table

4.3.4 CONTACT:


EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Home › Contacts › Add contact

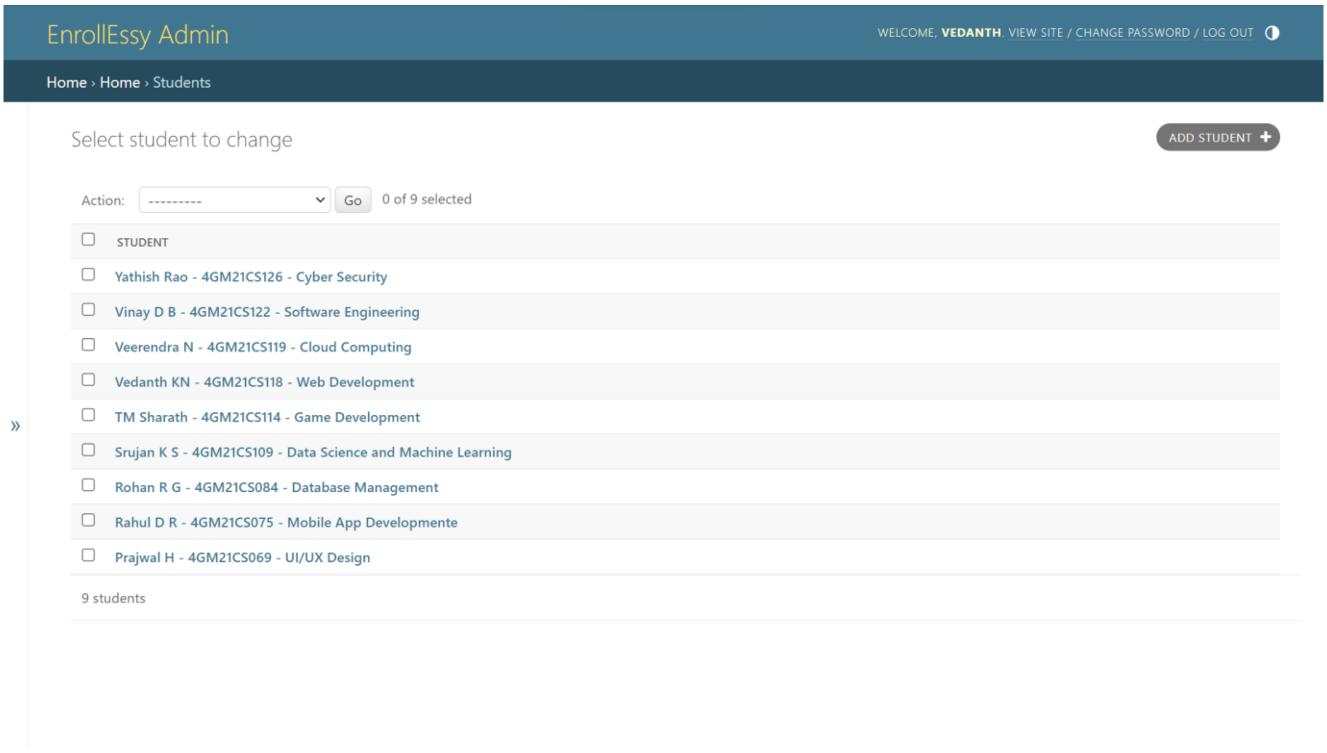
Add contact

Name:	
Email:	
Phone:	
Massege:	

Fig 4.6: Contact Table

4.4INSERTIONS:

4.4.1 STUDENT:



EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Home › Students

Select student to change

ADD STUDENT

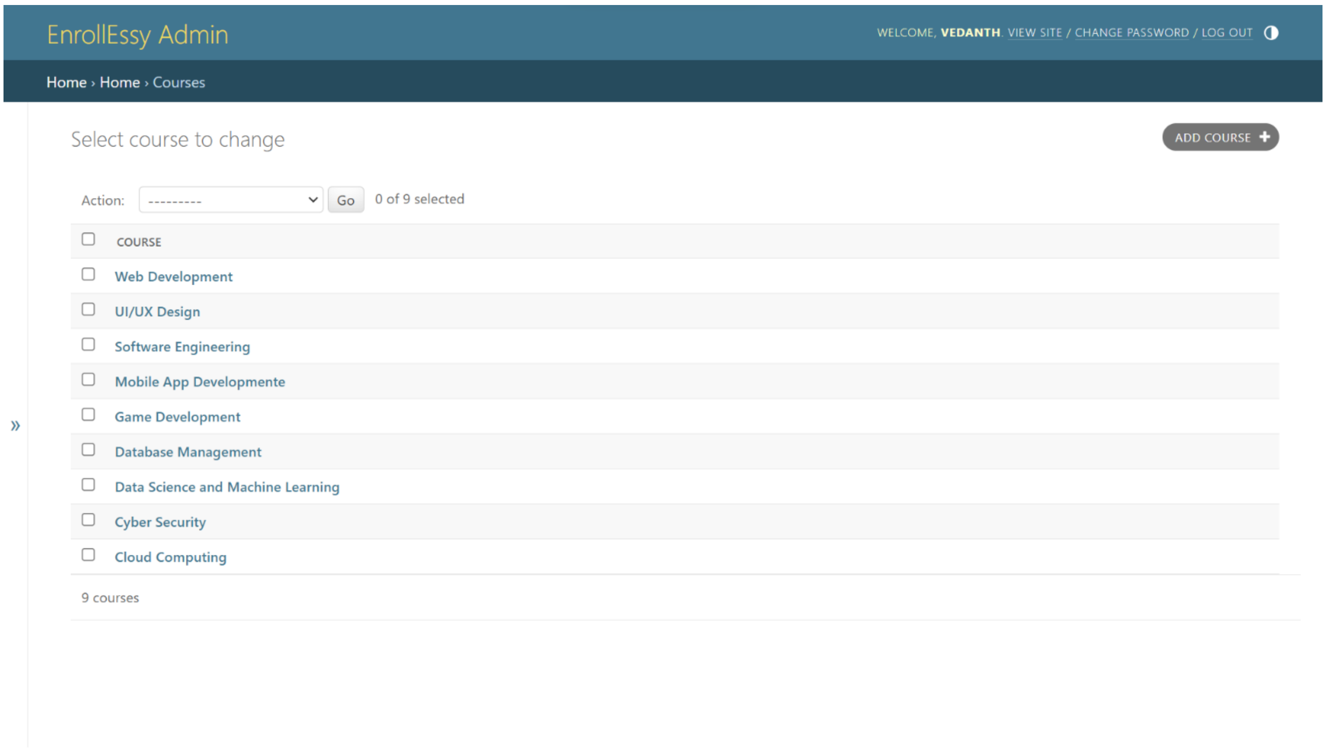
Action: --------- Go 0 of 9 selected

STUDENT
Yathish Rao - 4GM21CS126 - Cyber Security
Vinay D B - 4GM21CS122 - Software Engineering
Veerendra N - 4GM21CS119 - Cloud Computing
Vedanth KN - 4GM21CS118 - Web Development
TM Sharath - 4GM21CS114 - Game Development
Srujan K S - 4GM21CS109 - Data Science and Machine Learning
Rohan R G - 4GM21CS084 - Database Management
Rahul D R - 4GM21CS075 - Mobile App Developmente
Prajwal H - 4GM21CS069 - UI/UX Design

9 students

Fig 4.7: Student Insertions

4.4.2 COURSE



EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Home › Courses

Select course to change

ADD COURSE

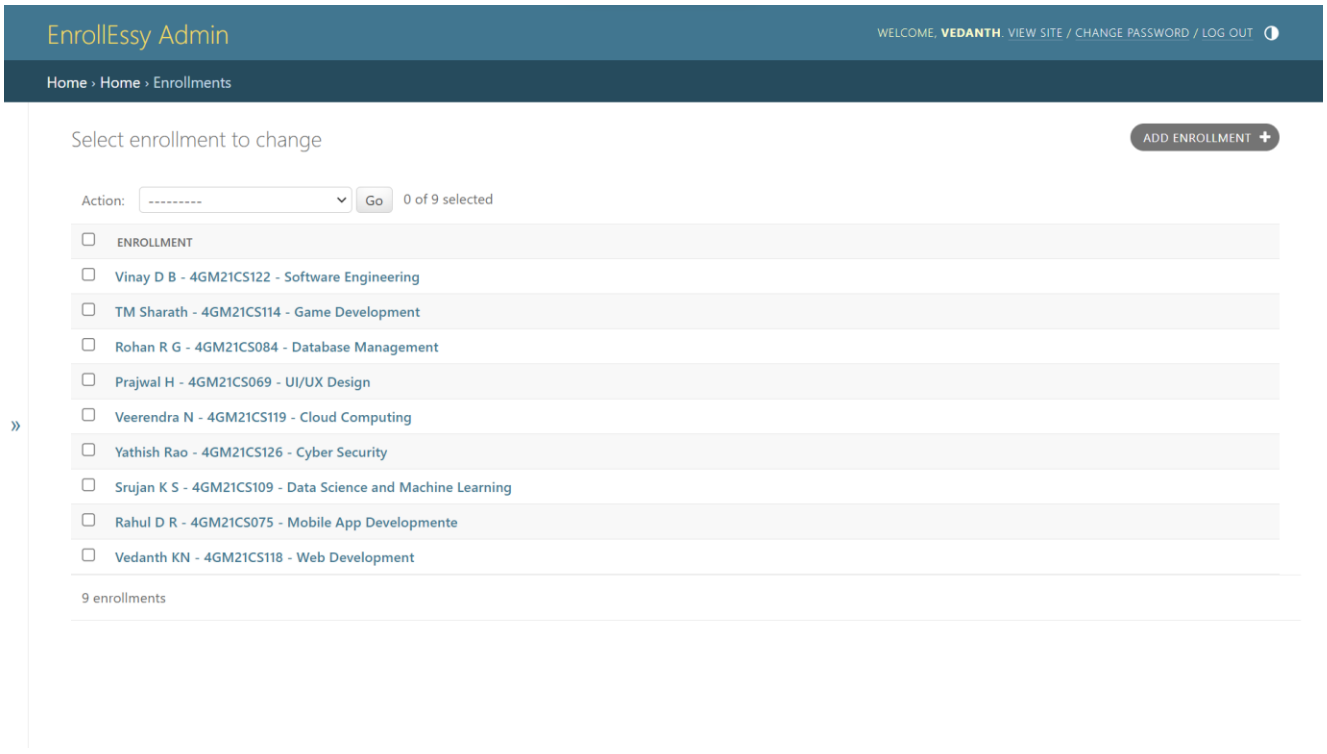
Action: --------- Go 0 of 9 selected

COURSE
Web Development
UI/UX Design
Software Engineering
Mobile App Developmente
Game Development
Database Management
Data Science and Machine Learning
Cyber Security
Cloud Computing

9 courses

Fig 4.8: Course Insertions

4.4.3 ENROLLMENT:



EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Home › Enrollments

Select enrollment to change

ADD ENROLLMENT

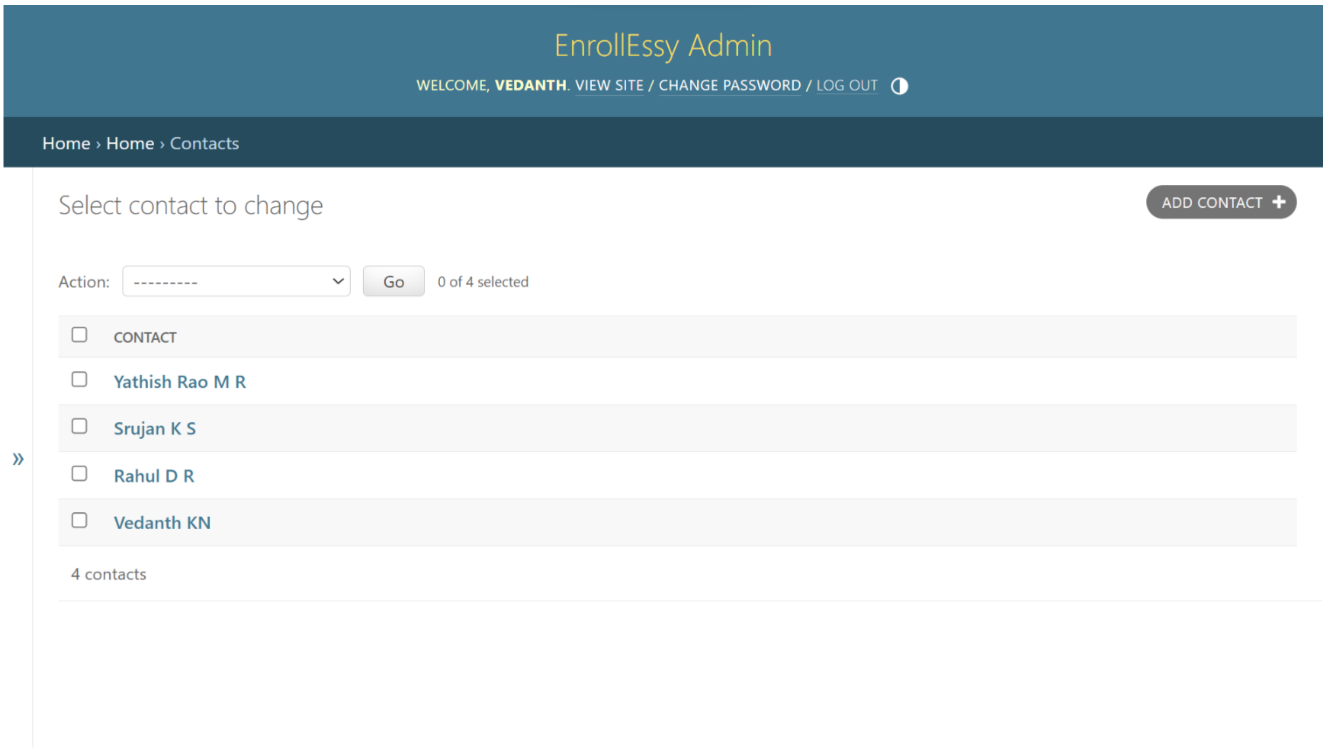
Action: --------- Go 0 of 9 selected

ENROLLMENT
Vinay D B - 4GM21CS122 - Software Engineering
TM Sharath - 4GM21CS114 - Game Development
Rohan R G - 4GM21CS084 - Database Management
Prajwal H - 4GM21CS069 - UI/UX Design
Veerendra N - 4GM21CS119 - Cloud Computing
Yathish Rao - 4GM21CS126 - Cyber Security
Srujan K S - 4GM21CS109 - Data Science and Machine Learning
Rahul D R - 4GM21CS075 - Mobile App Developmente
Vedanth KN - 4GM21CS118 - Web Development

9 enrollments

Fig 4.9: Enrollment Insertions

4.4.4 CONTACT:



EnrollEssy Admin

WELCOME, **VEDANTH**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Home › Contacts

Select contact to change

ADD CONTACT

Action: --------- Go 0 of 4 selected

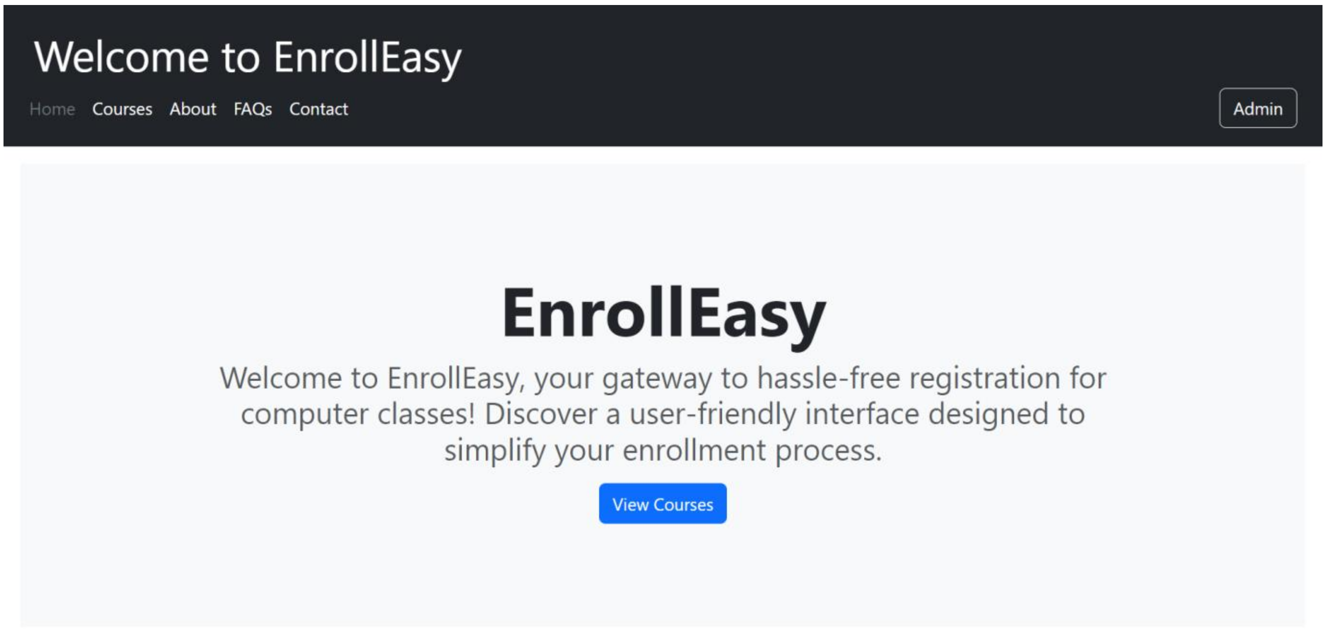
CONTACT
Yathish Rao M R
Srujan K S
Rahul D R
Vedanth KN

4 contacts

Fig 4.10: Contact Insertions

CHAPTER 5
WEBSITE SNAPSHOTS

5.1 HOME PAGE:

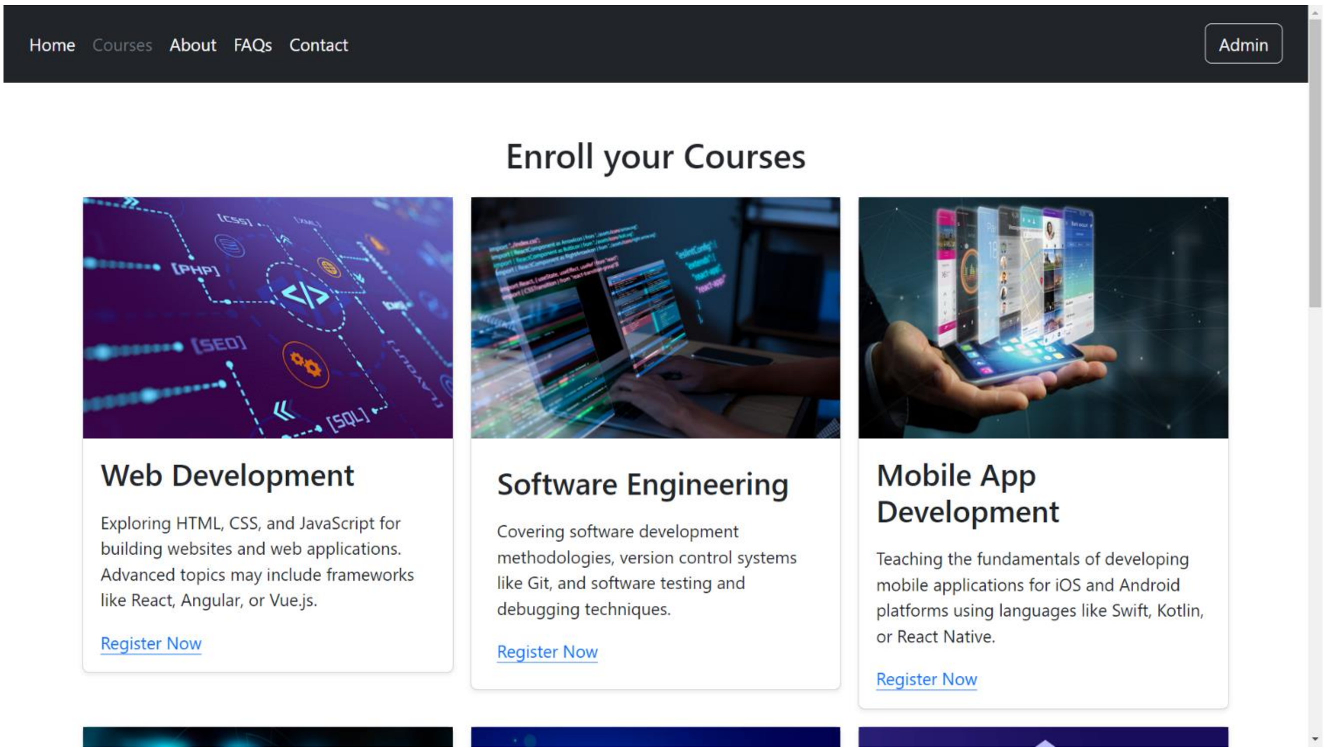


EnrollEasy

Welcome to EnrollEasy, your gateway to hassle-free registration for computer classes! Discover a user-friendly interface designed to simplify your enrollment process.

Fig 5.1: Home Page

5.2 COURESE PAGE:



Enroll your Courses

Web Development

Exploring HTML, CSS, and JavaScript for building websites and web applications. Advanced topics may include frameworks like React, Angular, or Vue.js.

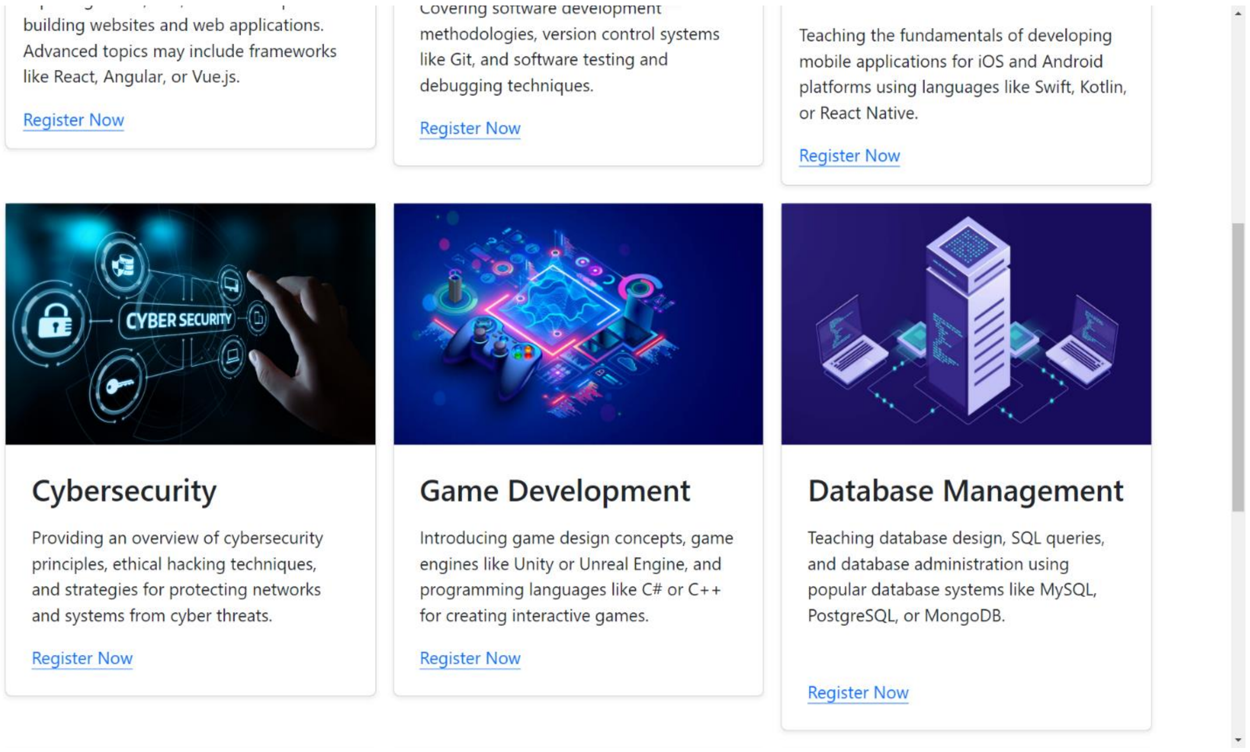
Software Engineering

Covering software development methodologies, version control systems like Git, and software testing and debugging techniques.

Mobile App Development

Teaching the fundamentals of developing mobile applications for iOS and Android platforms using languages like Swift, Kotlin, or React Native.

Fig 5.2: Course Page



building websites and web applications. Advanced topics may include frameworks like React, Angular, or Vue.js.

Register Now

Covering software development methodologies, version control systems like Git, and software testing and debugging techniques.

Register Now

Teaching the fundamentals of developing mobile applications for iOS and Android platforms using languages like Swift, Kotlin, or React Native.

Register Now

Cybersecurity

Providing an overview of cybersecurity principles, ethical hacking techniques, and strategies for protecting networks and systems from cyber threats.

Register Now

Game Development

Introducing game design concepts, game engines like Unity or Unreal Engine, and programming languages like C# or C++ for creating interactive games.

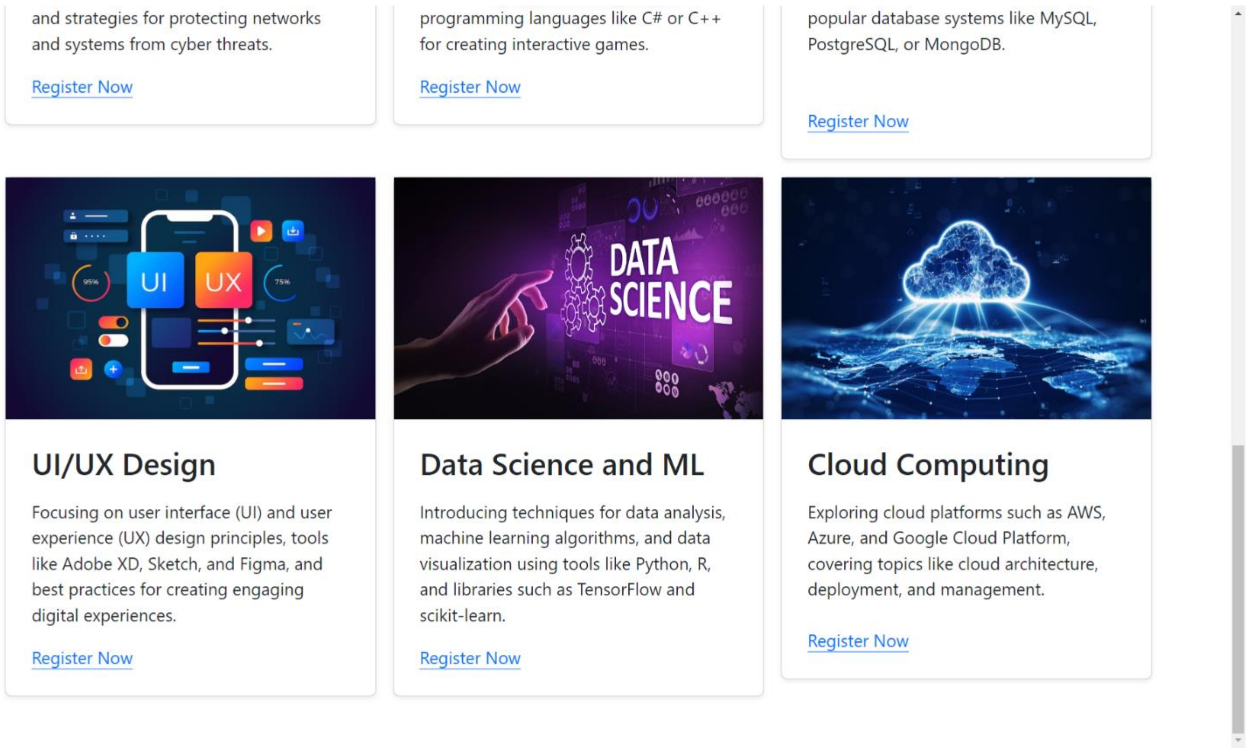
Register Now

Database Management

Teaching database design, SQL queries, and database administration using popular database systems like MySQL, PostgreSQL, or MongoDB.

Register Now

Fig 5.2.1: Course Page



and strategies for protecting networks and systems from cyber threats.

Register Now

programming languages like C# or C++ for creating interactive games.

Register Now

popular database systems like MySQL, PostgreSQL, or MongoDB.

Register Now

UI/UX Design

Focusing on user interface (UI) and user experience (UX) design principles, tools like Adobe XD, Sketch, and Figma, and best practices for creating engaging digital experiences.

Register Now

Data Science and ML

Introducing techniques for data analysis, machine learning algorithms, and data visualization using tools like Python, R, and libraries such as TensorFlow and scikit-learn.

Register Now

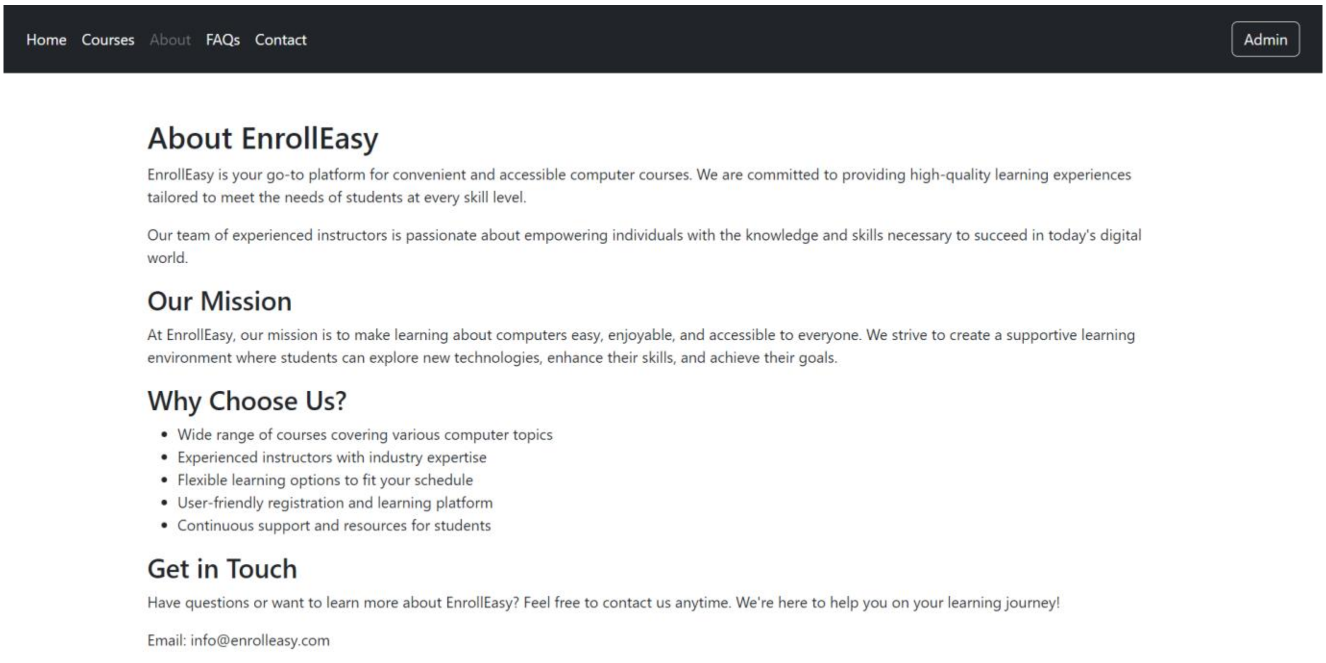
Cloud Computing

Exploring cloud platforms such as AWS, Azure, and Google Cloud Platform, covering topics like cloud architecture, deployment, and management.

Register Now

Fig 5.2.2: Course Page

5.3 ABOUT PAGE:



About EnrollEasy

EnrollEasy is your go-to platform for convenient and accessible computer courses. We are committed to providing high-quality learning experiences tailored to meet the needs of students at every skill level.

Our team of experienced instructors is passionate about empowering individuals with the knowledge and skills necessary to succeed in today's digital world.

Our Mission

At EnrollEasy, our mission is to make learning about computers easy, enjoyable, and accessible to everyone. We strive to create a supportive learning environment where students can explore new technologies, enhance their skills, and achieve their goals.

Why Choose Us?

- Wide range of courses covering various computer topics
- Experienced instructors with industry expertise
- Flexible learning options to fit your schedule
- User-friendly registration and learning platform
- Continuous support and resources for students

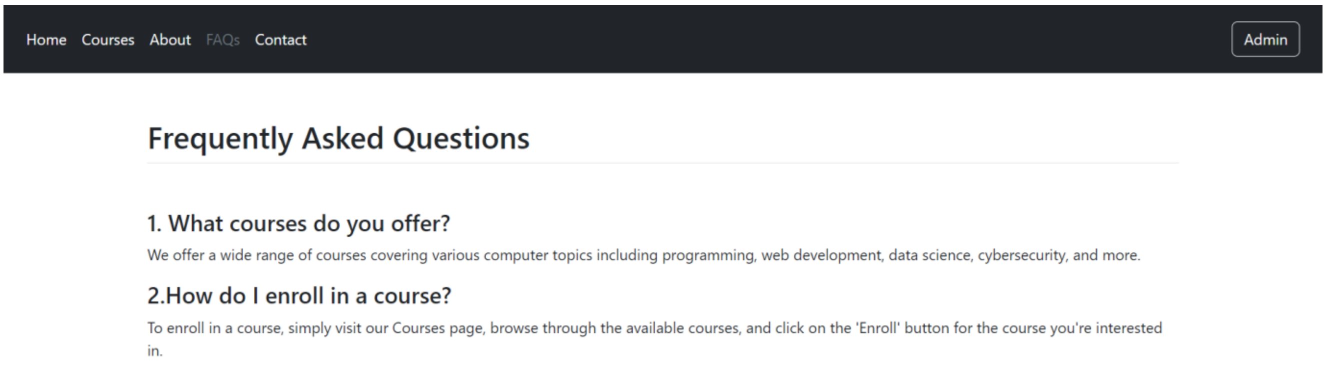
Get in Touch

Have questions or want to learn more about EnrollEasy? Feel free to contact us anytime. We're here to help you on your learning journey!

Email: info@enrolleasy.com

Fig 5.3: About Page

5.4 FAQS PAGE:



Frequently Asked Questions

1. What courses do you offer?

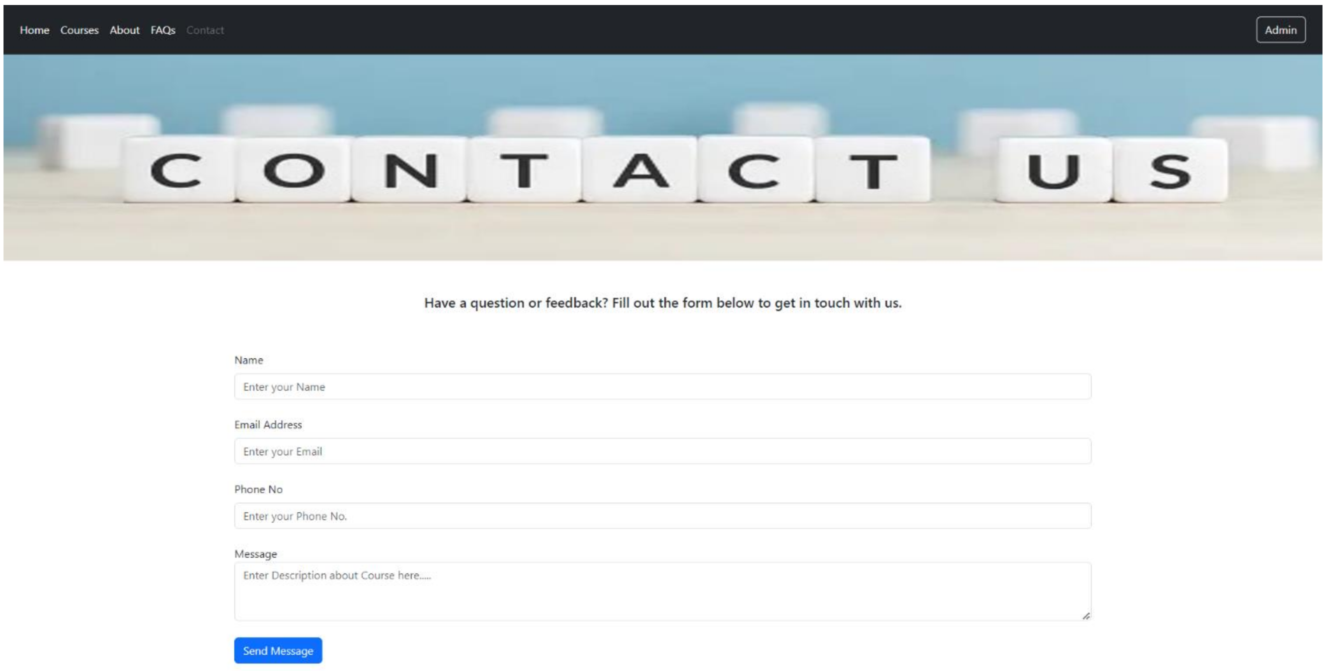
We offer a wide range of courses covering various computer topics including programming, web development, data science, cybersecurity, and more.

2.How do I enroll in a course?

To enroll in a course, simply visit our Courses page, browse through the available courses, and click on the 'Enroll' button for the course you're interested in.

Fig 5.4: FAQs Page

5.5 CONTACT PAGE:



Have a question or feedback? Fill out the form below to get in touch with us.

Name: Enter your Name

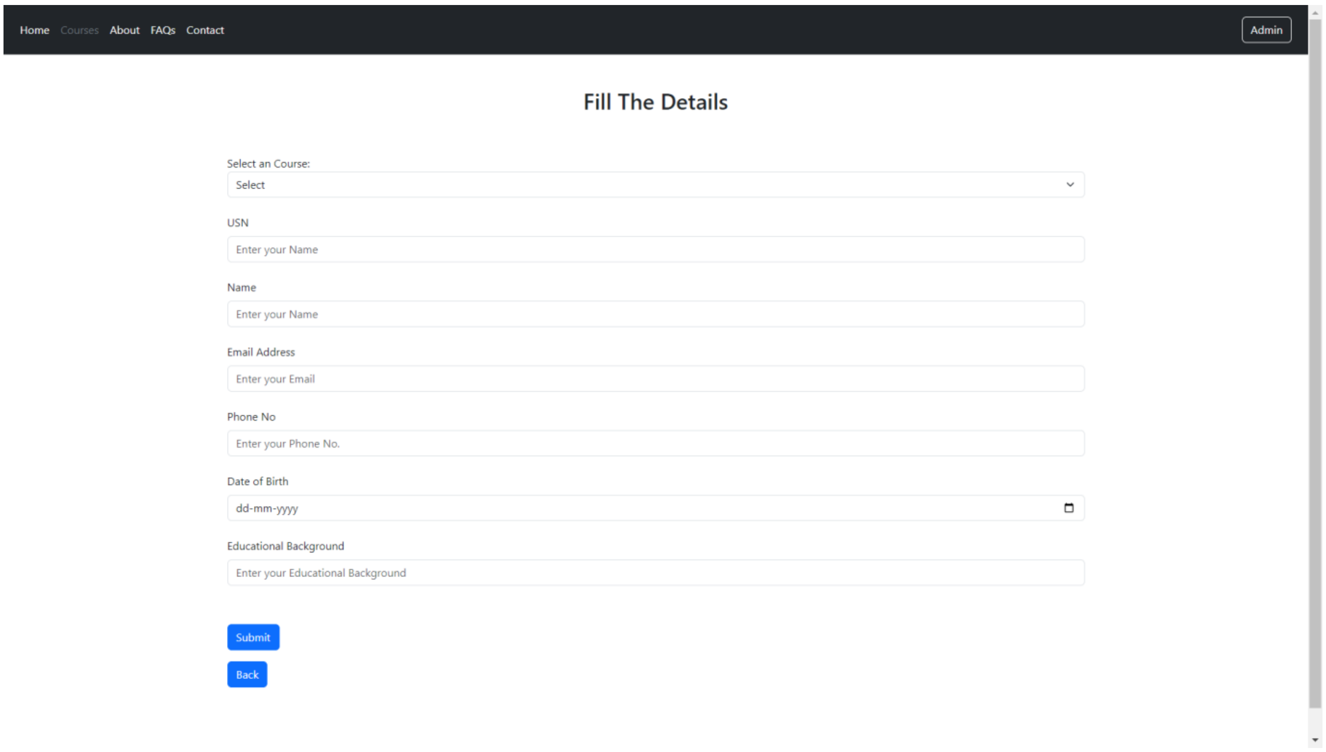
Email Address: Enter your Email

Phone No: Enter your Phone No.

Message: Enter Description about Course here.....

Fig 5.5: Contact Page

5.6 STUDENT PAGE:



Fill The Details

Select an Course: Select

USN: Enter your Name

Name: Enter your Name

Email Address: Enter your Email

Phone No: Enter your Phone No.

Date of Birth: dd-mm-yyyy

Educational Background: Enter your Educational Background

Fig 5.6: Student Page

CHAPTER 6
CONCLUSION

- **Enhanced User Experience:** The system aims to streamline the registration process for computer classes, offering a user-friendly interface for students to browse available courses, register for classes, and manage their profiles. This focus on user experience contributes to higher user satisfaction and engagement.

- **Efficient Course Management:** With features such as course categorization, enrollment tracking, and instructor management, the system provides administrators with efficient tools to manage courses, instructors, and student enrolments. This helps optimize resource allocation and improve course offerings based on demand.

- **Data Integrity and Security:** By leveraging Django's built-in security features and robust database management capabilities, the system ensures data integrity and protection against unauthorized access. Secure user authentication, data encryption, and role-based access control mechanisms contribute to maintaining the confidentiality and integrity of student and course data.

- **Scalability and Flexibility:** The system architecture, built on Python and Django, is designed to scale efficiently to accommodate increasing numbers of users, courses, and administrative functionalities. Modular design and adherence to best practices enable seamless integration of new features and enhancements as the system evolves.

- **Insightful Reporting and Analytics:** Through the integration of reporting and analytics features, the system provides administrators with valuable insights into enrollment trends, course popularity, student demographics, and other key metrics. This data-driven approach enables informed decision-making and strategic planning for future course offerings and resource allocation.

- **Community Engagement and Support:** The open-source nature of Python and Django fosters a vibrant developer community, offering access to a wealth of resources, tutorials, and third-party libraries. This ecosystem facilitates collaboration, knowledge sharing, and ongoing support for the project, ensuring its long-term sustainability and success.

REFERENCES

- https://getbootstrap.com/docs/5.3/examples/
- https://docs.djangoproject.com/en/5.0/topics/db/models/
- https://www.youtube.com/watch?v=JxzZxdht-XY
- https://chat.openai.com/