KLE Society's
KLE Technological University, Hubballi.
2022 - 2023

A Capstone Project Report

on

Question Paper Generation for Outcome Based Education

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by

SEEMA G AGER	01FE19BCS097
VIRUPAKSHA B M	01FE20BCS425
MUKTI BHANSALI	01FE19BCS203
SHIVAM KUMAR RAI	01FE19BCS259

Under the guidance of
Dr. Sujata C

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Hubballi – 580 031

2022 -23

KLE Society's
KLE Technological University, Hubballi.

2022 - 2023

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Capstone Project titled "Question Paper Generation for Outcome Based Education" is a bonafied work carried out by the student team comprising of Seema G Ager (01fe19bcs097), Virupaksha B M (01fe20bcs425), Mukti Bhansali (01fe19bcs203), Shivam Kumar Rai (01fe19bcs259) for partial fulfillment of completion of eighth semester B.E. in Computer Science and Engineering during the academic year 2022-23. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said program.

Guide(s) — Head, SoCSE

Dr. Sujata C — Dr. Meena S. M

Viva -Voce:

Name of the Examiners — Signature with date

1.

2.

Acknowledgement

We would like to thank our faculty and management for their professional guidance towards the completion of the project work. We take this opportunity to thank Dr. Ashok Shettar, Vice-Chancellor, Dr. N.H Ayachit, Registrar, and Dr. P.G Tewari, Dean Academics, KLE Technological University, Hubballi, for their vision and support.

We also take this opportunity to thank Dr. Meena S. M, Professor and Head, SoCSE for having provided us with direction and facilitated for enhancement of skills and academic growth.

We thank our guide Dr. Sujatha C, Professor, SoCSE for the constant guidance during interaction and reviews.

We extend our acknowledgement to the reviewers for critical suggestions and inputs. We also thank Project Co-ordinator Dr. Sujatha C. and faculty in-charges for their support during the course of completion.

We express gratitude to our beloved parents for constant encouragement and support.

Virupaksha B M - 01FE20BCS425
Seema G Ager - 01FE19BCS097
Mukti Bhansali - 01FE19BCS203
Shivam Kumar Rai - 01FE19BCS259

ABSTRACT

With the increase in technology, new courses and advanced knowledge of old courses are seeking a high demand. As a result, exam preparation has always been a topic of interest because exams play a crucial role in evaluating an individual's performance. And for this reason, it's crucial to have an intelligent system that monitors students' development in addition to their learning abilities, keeping a lid on their performance. In accordance with the requirements for the university's assessment, different faculties produce various question papers. Making question papers with multiple questions that meet the course's learning objectives is extremely challenging for teachers. For this, we provide a quick, simplified, random, and secure question paper creation method based on outcome-based education (OBE). Every action executed task by the system must be automated, so that storage space, bias, and security are not a challenge anymore. As the question bank continues to expand, it must automatically generate a variety of sets of questions from time to time without worrying about duplicates from the initial exam.

Keywords : *Bloom's Taxonomy(BL), Course Outcome(CO), Performance Indicator(PI), Topic Learning Outcome(TLO), Outcome-Based Education(OBE)*

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Exams play a crucial role in assessing students educational progress in today's competitive world, and the information age's technology is now replaced by productive applications of technology. Acceptable and preferred is any product that accurately reduces time and power consumption. As a result, creating software from data is an essential task. The instructor intends to produce significant documents per the autonomous university's guidelines and assessments in all academic courses that reject various tests. For teachers, managing all the course aspects while avoiding the repetition of questions from later estimates is far more difficult. Because there is no standardized procedure, the paper quality associated with this issue is anticipated to vary depending on the qualifications and teacher questions.

The problem of defining courses, semesters, syllabuses, and patterns is made even worse by a lack of qualified teachers. All of these things might also make the question paper less good at times. The specialist says a decent survey is the right blend of subjects (questions) directed by different boundaries: cognitive level, questionnaire difficulty, and distribution of scores. It is hard to make a good questionnaire with a lot of questions about the purpose of a course in terms of content and cognitive level. Thus, we are introducing a Programmed Question Paper Generator Framework which could decrease time admission by supplanting the traditional methodology of question paper generation. There are arrangements to enter and alter information appropriate to any educational associations with complete liberty.

A typical method for resolving the constraint satisfaction problem (CSP) is automatic test paper generation, which uses questions selected from the question bank to automatically generate various previously stated, this system's primary goals. Additionally, the system employs security measures that prevent question paper duplication. This empowers an instructive establishment to produce questions guaranteeing security and non-dullness of inquiry paper and is a help for associations with restricted staff and assets. For all of its tasks, our system provides fast operations, secure data storage, and high-sec paperwork

1.1 Literature Review / Survey

[1] "Automatic Question Paper Generator System"

In this paper, they have developed an Automatic Question Paper Generator System. This is a clever method of producing test questions for colleges and universities. This application's major goal is to automate the process and use less labor. Both the amount of time and the amount of data entry needed are kept to a minimum. As previously stated, this system's primary goals are to offer high security and data accuracy. As a result, the system is more effective and provides better service. They employed a heuristic approach to generate timetabling, which produces a number of excellent choices. The algorithm will provide an even number of classes without any conflicts. It eliminated manual labor and paperwork, which decreased complexity and timing. Both suggested a model that uses a probability for question paper non-repetition at the time the question paper is generated. The system accepts questions of several categories, including knowledge-based, memory-based, logic-based, and application-based questions. The primary goal of the algorithm is to prevent question repetition by randomization of the questions. According to their respective tasks, the administrator and subordinates have divided this work. The three groups—Administrator, Data Entry User, and Paper Generator—that make up the automatic question paper generator system they suggested are implemented with Java. This system has numerous benefits, including the fact that updating the data is a simple operation because it removes the need for human labor and produces test questions and answers in a matter of seconds. Since there is a task division between the administrator and subordinates, access to all of the contained resources is controlled. Both user and admin passwords are stored using algorithms for security reasons, which improves security. Even for non-repetitive nature, algorithms are utilized; for example, the shuffling method, which further employs the randomization algorithm, and the flag system to indicate certain queries. The main benefit of this system is its role-based hierarchy, in which administrators enter data and create test questions, while data entry users can only enter data but cannot create test questions, and paper generators can only enter data to create test questions. Despite all the benefits, the template incorporates all logins in a single window, which can be very confusing. As a result, just two hierarchies—admin and user—are supplied in this system, and the tasks are distributed among them.

[2] " Automatic Question Paper Generator Using Bloom's Taxonomy"

In this paper, a proposed automated design methodology for question paper generation that is implemented as a real-time application is presented. By allowing restricted access to their resources, the suggested study explains an automated system that illustrates the transition from the conventional way of paper generation to an automated process. Understanding users and their unique responsibilities within the institute can help with this. They used an effective technique that is entirely random and limits the recurrence of questions in test

papers after taking into account the significance of randomization in the process of paper generation. By comparing their individual responsibilities, administrators and subordinates can be distinguished. As a result, the automated system model for question paper creation that is created offers the advancement in terms of regulated access to resources, random question paper generation, and a completely secure, independent platform. Teachers can use their system to automatically generate test questions from the question bank, which is a great resource. Although the system They developed stands out among other systems, there is room for further improvements to increase its utility. For instance, the system can be programmed to choose particular question kinds based on the sort of evaluation required. For instance, it would be shrewd to include all MCQs if the user requested an assessment for an online quiz. also, if a user is selecting. Short response questions, more objective types of questions, and term test assessments should be chosen. Users would also be thrilled to have a function that could offer data regarding discrepancies between user-provided specifications and system-generated specifications. The system currently just generates question papers, but in the future, it may even be incorporated with separate student logins for online test assessment with randomly produced questions at that specific time, making it more effective for Exam conduction. Currently, the system only generates question papers.

[3] "An Intelligent Question Paper Generator Using Randomized Algorithm"

In this paper, the proposed software is able to satisfy the needs of institutions in order to generate question papers in pdf or doc files. Text parsing methods are used to process the Layout of question paper to identify section categorization. Most of the work is done in the English language and generates Theoretical type questions. The automatic question generation system is an open area where still there is a scope of research is there for proposing methodologies by identifying complexities and types of questions like one-word answers and True or False needs generated. The aim of Qbase is to provide a platform to the user working for any educational institute. The user of Qbase can generate question papers according to their specific need. It will allow users to live edit, and print question papers and also it can be shared to multiplatform.

[4] "QBase – A Bloom Taxonomy-based Question Paper Generator"

In this paper, they proposed a Python-based desktop application for automatically generating question papers. This system will be able to generate question papers based on the skeleton of question paper which the user has to decide. The total marks are to be given as input and system will automatically divide these marks into questions which are also to be described by the person who is generating them. For developing this system we used scikit-fuzzy package of python for “fuzzy logic” algorithm. This system is more reliable in terms of duplicity removal, compromise issues, and security.

[5] "Adaptive Question Paper Generation System"

In this paper, they implemented the requirement model for OBE Oriented Auto-Generator

Of Examination Question (AGEQ). AGEQ applied a genetic algorithm to resolve issues around generating examination questions based on OBE specification. The tool helps to support the lecturer's effort in the construction of the examination paper and at the same time validates the process with OBE specification. Future works on this subject include looking into optimization of the objective function and fine-tuning GA parameter values to improve performance.

[6] "Designing GA-Based Auto-Generator of Examination Questions"

In this paper, they implemented the Question Database or Question Bank using a graphical hierarchy with the integration of the concept map of that curriculum which stores questions pertaining to a specific curriculum. A user Database is a relational database that stores information of the users, authorized to interact with the system i.e. QPD and QD. Question Paper Database is a relational database which stores the Question Papers generated by QPDE and is accepted by the QPD for the future review.

[7] "Design of Adaptive Question Bank Development and Management System"

In this paper, they have designed an adaptive question bank management system that is intelligently picking questions from the rich database (question bank) and representing the question model according to the inputs or parameters provided by the question paper designer (QPD). The system is using graphs for constructing concept map and a database for storing questions.

[8] "Fuzzy Logic-Based Intelligent Question Paper Generator"

In this paper, a fuzzy logic-based model is proposed for autonomous paper generation, using MATLAB. Comparative analysis with the classical method is done and fuzzy model is found to be more reliable, fast, and logical. The fuzzy logic-based approach is implemented for logical selection of these parameters while framing question paper for every subject irrespective of its discipline.

[9] "Prototype deployment of examination and quiz system (EQs) with outcome-based education in networking system"

In this paper, the EQs met the criteria for creating a system of tests and quizzes that was successful and efficient and was based on OBE. A web-based tool called EQs assists coordinators in creating quiz questions and lecturers in creating quizzes. Additionally, students can complete the quiz through this online system, which allows the lecturer to automatically find out how well the students performed in a particular subject during the exam session. On the other hand, the test questions can be directly converted by the coordinator into the final exam question paper for quicker and more precise activities. Additionally, EQs give users (lecturers and coordinators) the ability to create necessary documents, such as the Table of Specification Tests document, the final exam question paper, the student's fuzzyrmance document, and the schema answer for the final exam. A reliable algorithm for improved performance evaluation must be created for future work. To stop information leaking, the EQs also need to have, an anti-hackinThe fuzzy-logic-based in.

[10] "Android-based exam paper generator (Android-based E-PAGE)"

In this paper, they created an Android-based exam paper generator in this article. This generator creates exam papers based on specification toutcome-baseda variety of criteria and a strict time limit, questions from the question bank are chosen. This method uses an Android-based application to generate the model question paper with just a few clicks. When creating the exam paper, randomization and non-redundancy are also taken into account. As a result, the Android-based system that results is significantly more secure, randomised, non-redundant, and optimised.

[11] "Question paper generator and answer verifier"

In this paper, they have developed software that allows students to submit assignments, have them corrected through the software, and have their grades automatically updated in the college database. The instructions are given to the person who created the assignment. The system will create a dataset of all the questions that satisfy all the restrictions based on the settings chosen by the user. From this dataset, 5 questions will be randomly chosen and included in the assignment. This system's output has a big impact. It offers a high rate of accuracy for checking papers. The existing system can be expanded with a variety of new modules, making it even more flexible in terms of how it may handle data that needs to be corrected. As an illustration, consider adding a module that would accept voice data from a microphone and fix it automatically using the same technique. Additionally, a module can be added to accept responses from an image source, allowing for the scanning and correction of entire papers using algorithms for converting images to text.

[12] "Automatic Generation of Question Paper from User Entered Specifications Using a Semantically Tagged Question Repository,"

In this paper, they have developed an Automatic Generation of a Question Paper system. It was discovered by using this approach that the accuracy in terms of question selection is 95 percent. Five professors with at least five years of experience in computer science evaluated the usability and user-friendliness. "Explore each component of the system," "Validate each entry on every screen with positive and negative inputs," and "Check the accuracy of generated question paper in XML format and word format" were the simple directions provided to users during the initial round of user testing. The technology is a terrific tool for teachers to automatically generate test questions from category techniques and repositories. Although the system we created stands out among other systems, there is still room for improvement to increase its utility. The system may be programmed to choose various question types depending on the sort of evaluation required. For instance, it would be wise to include all MCQs if the user wanted to be assessed for an online quiz. Alternatively, more objective-style and short-answer questions must be preferred if the user opts for term test assessment. Users would also be thrilled to have a tool that could display data regarding discrepancies between user-provided requirements and system-generated specs.

1.2 Motivation

- The traditional method of question paper generation is time-consuming and hectic.
- It is very challenging for the teachers to cover all aspects of the course objectives and avoid duplication of questions in the subsequent exams.
- There are no standardized methods and hence the quality of the question paper depends completely on an individual teacher's experience and expertise.
- It requires going through vast techniques and picking questions that require a lot of time and employee work. Sometimes there is a need to generate question papers at the eleventh hour.
- The fact that there is a shortage of experienced teachers makes the situation even worse. At times, all these factors may deteriorate the quality of the question paper.
- Making a good question paper containing varied questions aligned with the learning objective of the course in terms of contents and cognitive level is very difficult.

1.3 Problem Statement

To develop a web application for faculty to generate question papers based on Outcome-Based Education (OBE).

1.4 Applications

- To generate question paper based on the user prompts.
- Preparing question paper for any course.
- Maintaining and keeping track of records of all generated question papers.
- It can be used in Universities to maintain their program structure.
- It allows to maintenance faculty database.

1.5 Objectives and Scope of the project

1.5.1 Objectives

- To design and create a database for the required application.
- To upload the question bank to the database.
- To create a question paper template.
- To select the questions as per the user template based on OBE.
- To view and download the generated question paper.

1.5.2 Scope of the project

In this project, an OBE-based question paper is implemented with several other functionalities. This software can keep records of the courses along with the branch and the program. It also maintains a database of the teaching and assistance staff. This software is designed in a structured manner to carry out the process of generating a question paper based on various parameters such as PI, CO, Bloom's level, and so on. It is University specific software that is designed for University purposes only.

Chapter 2

REQUIREMENT ANALYSIS

2.1 Functional Requirements

- The system shall be able to authenticate the user.
- The user shall be able to log in successfully to the system as admin or faculty.
- Admin shall be able to add a new program, department, course, or chapter.
- Admin shall be able to add, delete or modify faculty data.
- Faculty shall be able to add questions.
- Faculty shall be able to set question paper format.
- Faculty shall be able to retrieve questions based on PI, CO, and Bloom's taxonomy.
- The system shall be able to generate question papers based on user input.
- Faculty shall be able to download the generated question paper.

2.2 Non Functional Requirements

- The response of the website should be within 5 seconds.
- The format of the question paper should be .docx, .doc, or .pdf format.
- The website should be redirected to its other pages within 5 seconds.
- The login credentials of the data should be encrypted.

2.3 Software Requirements

2.3.1 MERN Stack

MERN stands for MongoDB, Express, React, and Node, after the four key technologies that make up the stack.

- MongoDB: document database
- Express(.js) : Node.js web framework
- React(.js) : a client-side JavaScript framework
- Node(.js): the premier JavaScript web server
- Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js is a popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

Working

The MERN architecture allows you to easily construct a three-tier architecture (front end, back end, database) entirely using JavaScript and JSON.

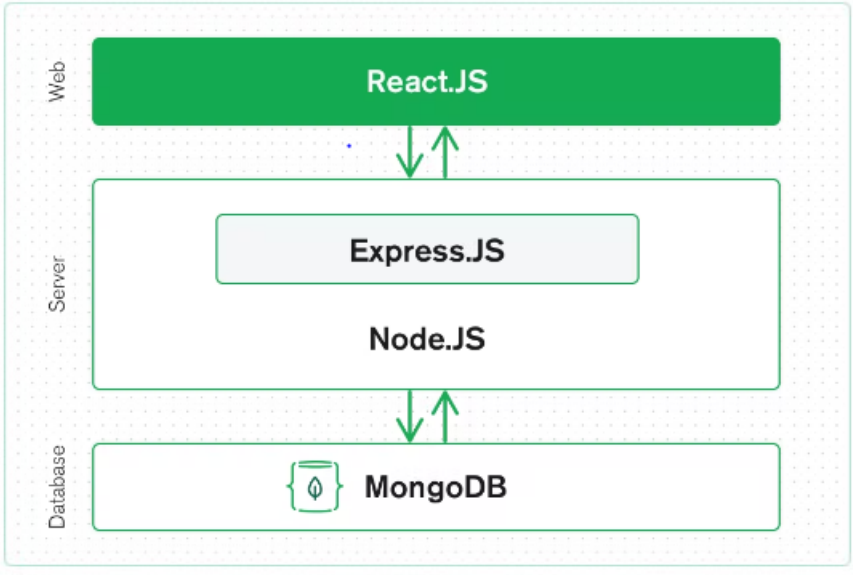


Figure 2.1: MERN Stack

React. js front end

React.js, a declarative JavaScript framework for building dynamic client-side apps in HTML, makes up the top tier of the MERN stack. React enables you to link simple components to data on your back-end server, connect complicated interfaces to those connections, and render those interfaces as HTML. React excels at handling stateful, data-driven interfaces with little effort and code, and it includes all the features you'd expect from a contemporary web framework, including excellent support for forms, error handling, events, lists, and more.

Express. js and Node.js server tier

The server-side framework Express.js, which functions inside a Node.js server, is the next level below. Express.js indeed describes itself as a "fast, unopinionated, minimalist web framework for Node.js." Express.js offers robust models for handling HTTP requests and responses as well as URL routing (correlating an incoming URL with a server function). You can connect to the Express.js functions that power your application by sending XML HTTP Requests (XHRs), GET requests, or POST requests from your React.js front end. These functions then access and update data in your MongoDB database using the Node.js drivers for MongoDB, either through callbacks or promises.

MongoDB database tier

If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you're going to want a database that's just as easy to work with as React, Express, and Node. That's where MongoDB comes in: JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval.

2.4 Database Requirements

2.4.1 ER Diagram

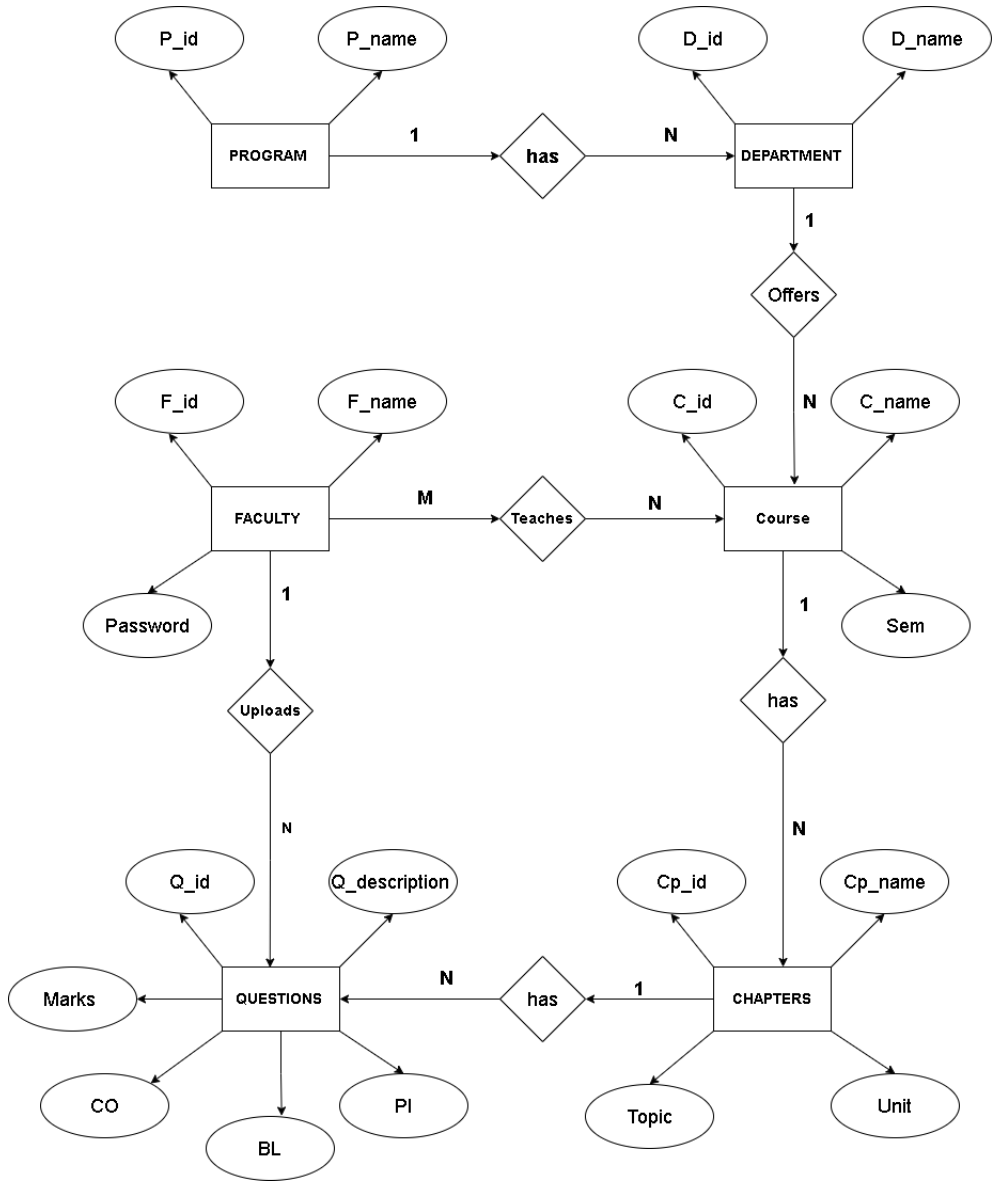


Figure 2.2: ER diagram

Figure 2.2 shows the ER diagram of the system. All the entities with relations are included in this diagram. The program entity contains the program id and the program name which has many departments. Each department consists of information on its id and the department name. A department can offer several courses where each course has its unique id, name, and number of credits allotted to it. Various courses are assigned to several faculties that have an id, name, and role. The course has chapters and the chapters in turn have questions. The

chapter details include chapter id, name, and unit number. The questions have attributes of id, marks, CO, BL, and PI. These questions are uploaded by the faculty.

2.4.2 Relational Schema

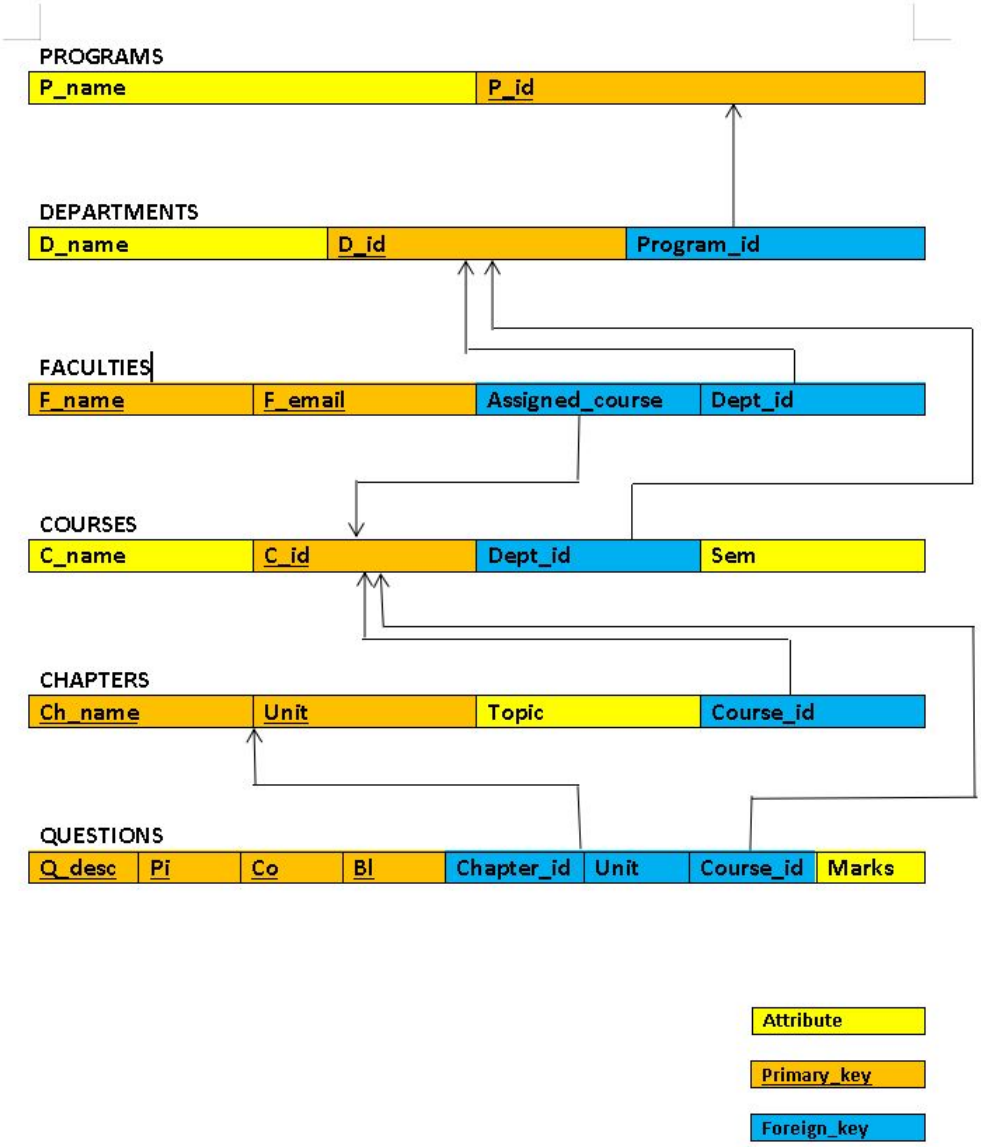


Figure 2.3: Relational Schema

Figure 2.3 shows the Relational schema of the system. It has tables named Programs(Pname,pid), Departments(Dname, Did, Pid), Faculties(Fname, Femail,Password,Did), Courses(Cname, Cid, Did, Sem), Chapters(Chname, Unit, Topic, Cid), and Questions(Qid, PI, CO, BL, Chid, Unit, Cid, Marks)

Chapter 3

SYSTEM DESIGN

3.1 Architecture Design

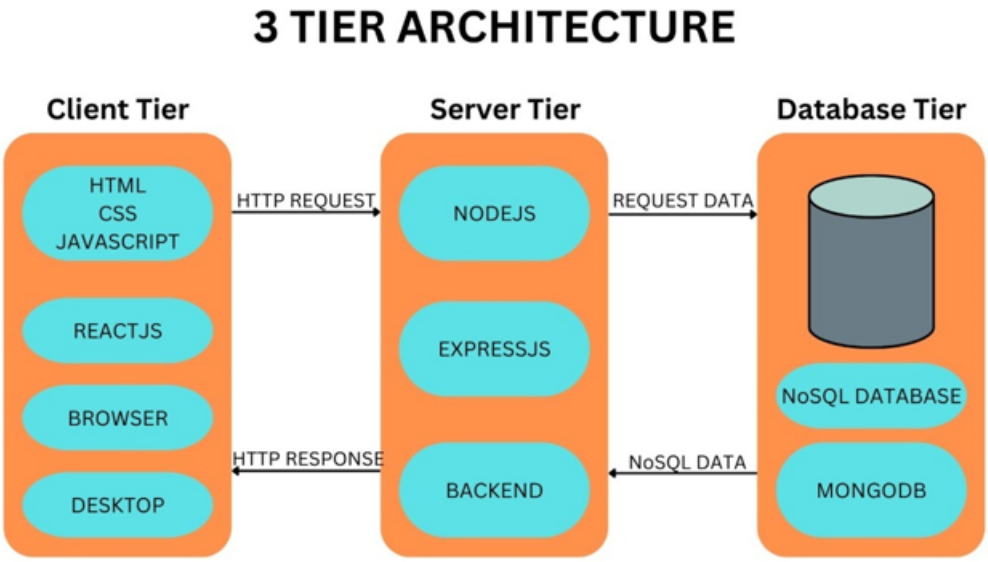


Figure 3.1: Client-Server Architecture

The client-server architecture has various components for distribution for the processing of requests. For this system, we define a 3-tier architecture consisting of a client, a server, and a database. The client tier consists of the front-end part of the system while the server and the database consist of the back-end part of the system. The client tier is built from the languages HTML, CSS, and Javascript and the modern technology ReactJs. This is viewed on the user's desktop via his browser. The server tier is built using the technologies NodeJs and ExpressJs. The database is a NoSQL database that is written using MongoDB database language. When the client sends an HTTP request, the server responds to the request. The server requests data from the database and the database responds to the server with the asked data. The server then responds to the client with the required information. This is the overview of the working of our system using a client-server architecture.

3.2 Work Flow

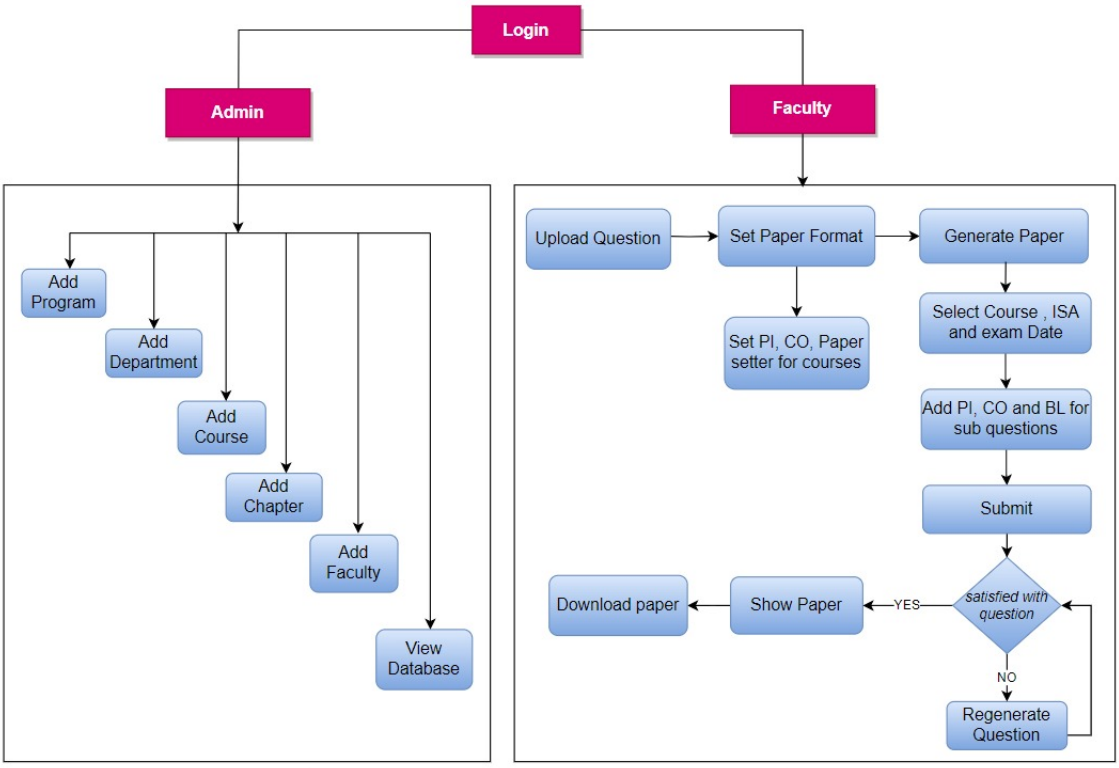


Figure 3.2: Control flow architecture

This is the control flow architecture of the system which defines the accessibility of various users along with the flow of information. The login function can be accessed by two kinds of users; Admin or Faculty. The admin has a set of functionalities like adding a program, adding departments, adding courses, adding chapters, adding faculty, and setting question paper skeleton. The faculty has a series of functionalities where to access one section, there is a pre-requisite of previous sections. To generate a question paper, the faculty also uploads a set of questions to balance the number of questions that appeared and did not appear in question papers. Then, the question paper pattern is selected after which the process of generating the question paper can be started. Here, the faculty enters the PIs, COs, and BLs of a main question and the algorithm then generates the sub-questions for the main question. The questions can be finalized or can be regenerated where the algorithm selects a different set of questions from the previous one. This process can be repeated until the final questions

are selected for the exam. When all the questions are selected for the main questions, the question paper can be generated and printed.

3.3 UML Diagram

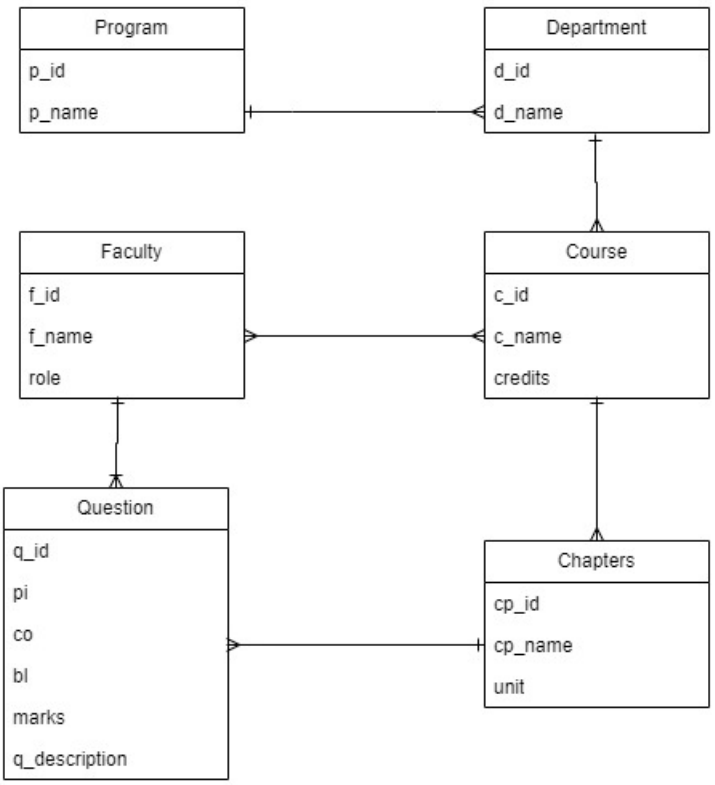


Figure 3.3: Class Diagram

This figure shows the UML diagram of the system. All the classes with ordinality are included in this diagram. The program entity contains the program id and the program name which has many departments. Each department consists of information on its id and the department name. A department can offer several courses where each course has its unique id, name, and number of credits allotted to it. Various courses are assigned to several faculties that have an id, name, and role. The course has chapters and the chapters in turn have questions. The chapter details include chapter id, name, and unit number. The questions have attributes of id, marks, CO, BL, and PI. These questions are uploaded by the faculty.

Chapter 4

IMPLEMENTATION

4.1 Working

The web application allows access to different functionalities based on admin and faculty login. Admin will have major database control such as adding, viewing, and deleting data to the database. Admin will have control over faculty registration, thus faculty added by admin can only be able to login to the website. Faculty will mainly perform functionalities related to question paper generation such as adding questions to the database for various courses and generating question papers for exams following the OBE pattern.

4.1.1 Faculty Login:

1. Enters email and Password. Gets Redirected to the homepage.
2. Can perform three functionalities

   i) Upload Questions

   ii) Set Paper Format

   iii) Generate Paper
3. Upload Question :

   - Select course, chapter, and topic.
   - Fill in the question details such as question description, PI code, CO(Course Outcome), BL(Bloom's Level), and marks.
   - Click on ADD button.
   - The Question will get added to the database.
4. Set Paper Format:

   - Select course and ISA.
   - Enter the Paper Setter email.
   - Add the PI code and CO(Course Outcome) to be included in the question paper while generating the paper.

- Click on Submit button.

5. Generate Paper :

   - Only the faculties who are allowed to generate paper will be able to perform this functionality for only those courses which contains the paper setter email same as the login email.
   - Select course - (if the faculty is allowed to generate a paper for the selected course then, Question Code Details will be shown).
   - Select ISA and Exam Date. Now, A Paper Details Form will appear after filling the above form if the Question code is displayed or else the user is not assigned as a paper setter for the selected course.
   - Three Question form will appear where PI code, CO, and BL are supposed to be filled and submitted to generate question Regenerate the section if not satisfied with it.
   - Finally click on Show Paper to view the preview of the question paper and then click on Download Paper and save it to your system.

4.1.2 Admin Login:

1. Enters admin credential and gets redirected to the admin page.
2. Can perform functionalities such as :

   - Add Programs
   - Add Departments
   - Add Courses
   - Add Chapters
   - Add faculties
   - View Database

3. Under View Database, admin can view faculty details, Program Details, Department Details, Course Details, and Chapter Details.

4.2 Algorithm

4.2.1 Algorithm for Question Generation

- If Number of sub-question = 3

  Approach: To generate all possible triplets and compare the sum of marks of all three arrays of questions with the total marks.

  Assumption : Total marks or X = 20;

  Algorithm :

  1. Given three arrays arr1, arr2, arr3 containing question details and a total mark X.
  2. Create three nested loops, first loop runs from start to end (loop counter i) for arr1, the second loop runs from I + 1 to end (loop counter j) for arr2, and the third loop runs from j+1 to end (loop counter k) for arr3.
  3. The counter of these loops represents the index of 3 questions of the triplets.
  4. Find the sum of marks of the ith question of arr1, jth question of arr2, and kth question of arr3. If the sum is equal to the total mark X, store the triplet of the question in an array.
  5. After traversal of all three arrays, display the generated array question one at a time using Randomization Algorithm.

  Time Complexity : $O(N^3)$

- If Number of sub-question = 2

  Approach: To generate all possible pairs and compare the sum of marks of both arrays with the total marks.

  Assumption: Total marks or X = 20;

  Algorithm :

  1. Given two arrays arr1 and arr2 containing question details.
  2. Create two nested loops, first loop runs from start to end (loop counter i) for arr1, and the second loop runs from I + 1 to end (loop counter j) for arr2.
  3. Find the sum of marks of the ith question of arr1 and the jth question of arr2. If the sum is equal to the total marks X, store the pair of questions in an array.
  4. After traversal of both arrays, display the generated array question one at a time using Randomization Algorithm.

  Time Complexity : $O(N^2)$

4.2.2 Randomization Algorithm

- The java script Math.random() method is used to return a floating point pseudo-random number between range [0, 1). This random number can be scaled to the desired range (Total size of the possible question array).
- The Math. random() method is used with the Math .floor() function to return random integers with a maximum value of the size of the generated question array.
- Algorithm used is called xorshift128+
- The numbers generated by xorshift128+ aren't random, the sequence just takes a long time to repeat and they're relatively evenly distributed over the expected range of values.

4.3 API Testing using Postman

Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple GUI for sending and viewing HTTP requests and responses.

4.3.1 Post Requests

The POST request is an HTTP method. We use this method when additional information needs to be sent to the server inside the body of the request. In general, when we submit a POST request, we expect to have some changes on the server, such as updating, deleting, or inserting.

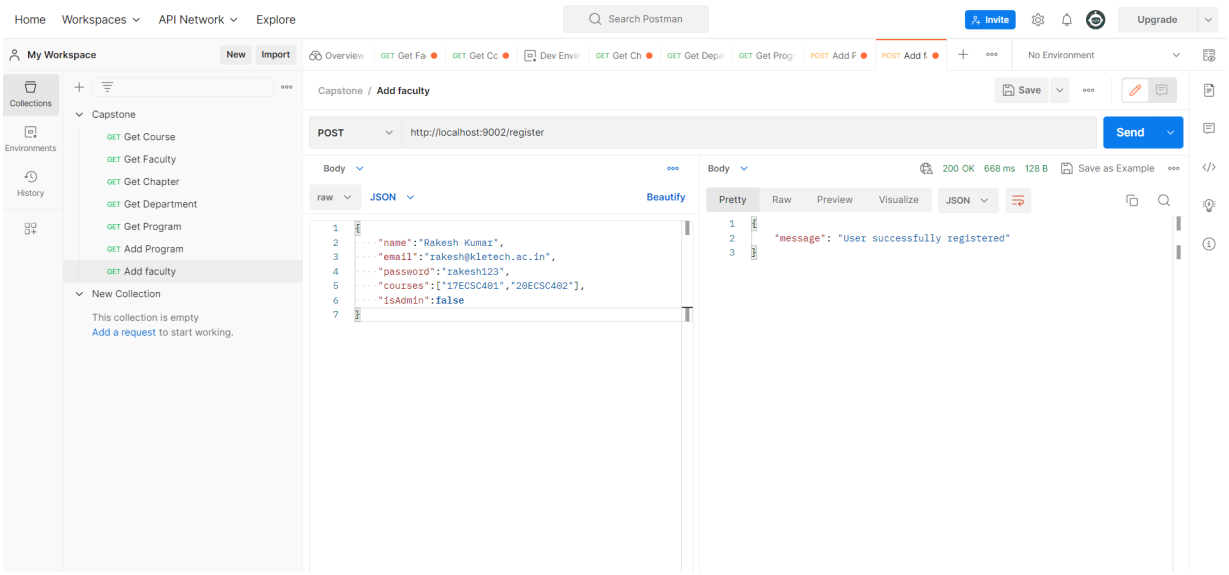


Figure 4.1: Add Faculty

A post request to the server for inserting faculty details which contains name, email, password and assigned courses, and isAdmin field. This request will insert the data in the faculty table in the database if the same email is not present in the database and will send a response with the status code “OK” along with the message “User successfully registered” or else it will send a message “User already exists”.

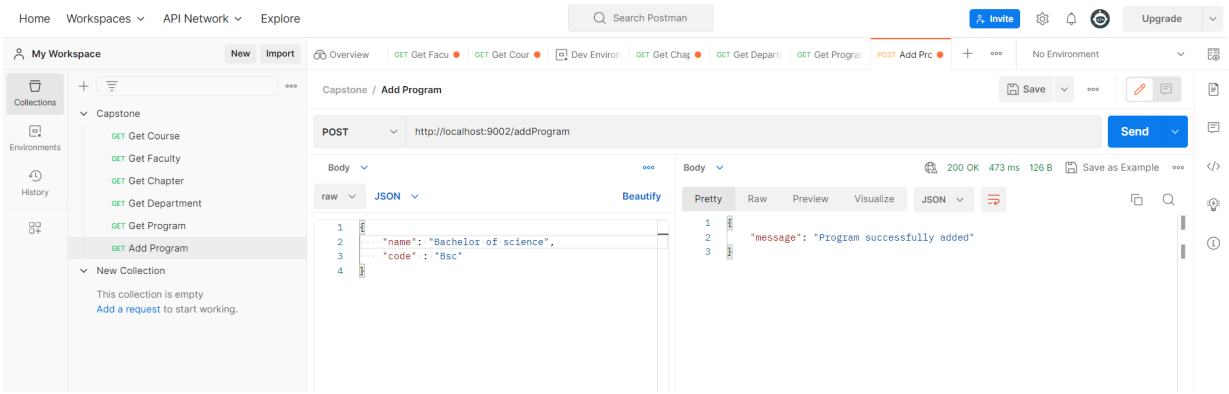


Figure 4.2: Add Program

A post request to the server for inserting program details which contains the program name and program code. This request will insert the data in the program table in the database if the same program code is not present in the database and will send a response with the status code “OK” along with the message “Program successfully added” Else it will send a message “Program already exist”.

4.3.2 Get Requests

A GET request gets the information from the server. When we make the GET request on the server, the server responds to the request.

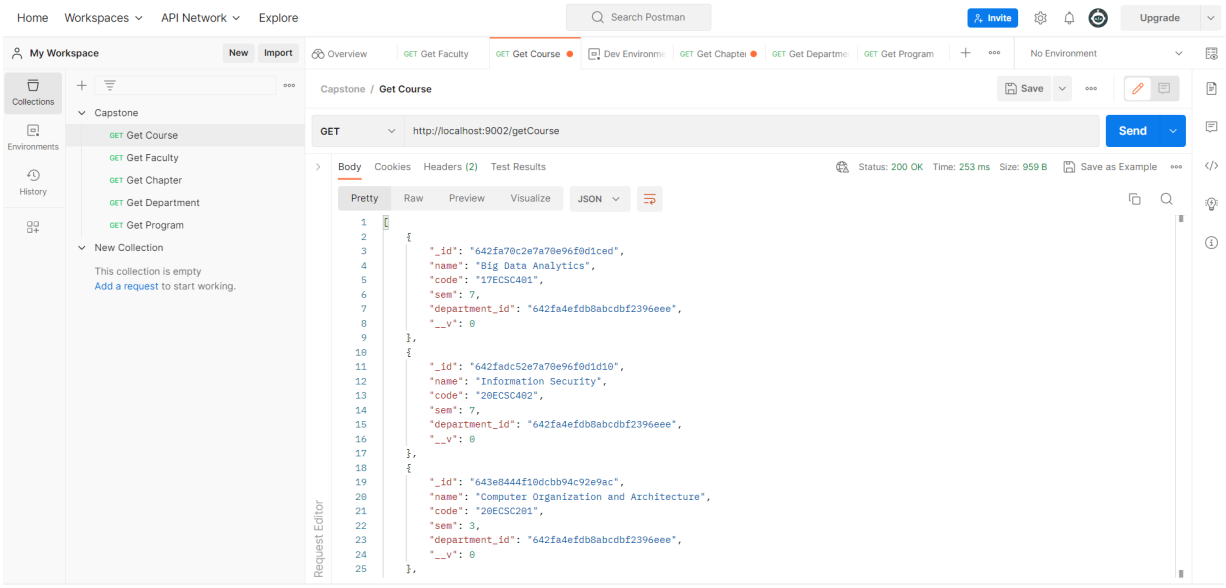


Figure 4.3: Get Course Details

A request to the server for retrieving the course details from the database. If the API is working fine, then it will send a response with the status code “OK” and will display all the retrieved course details.

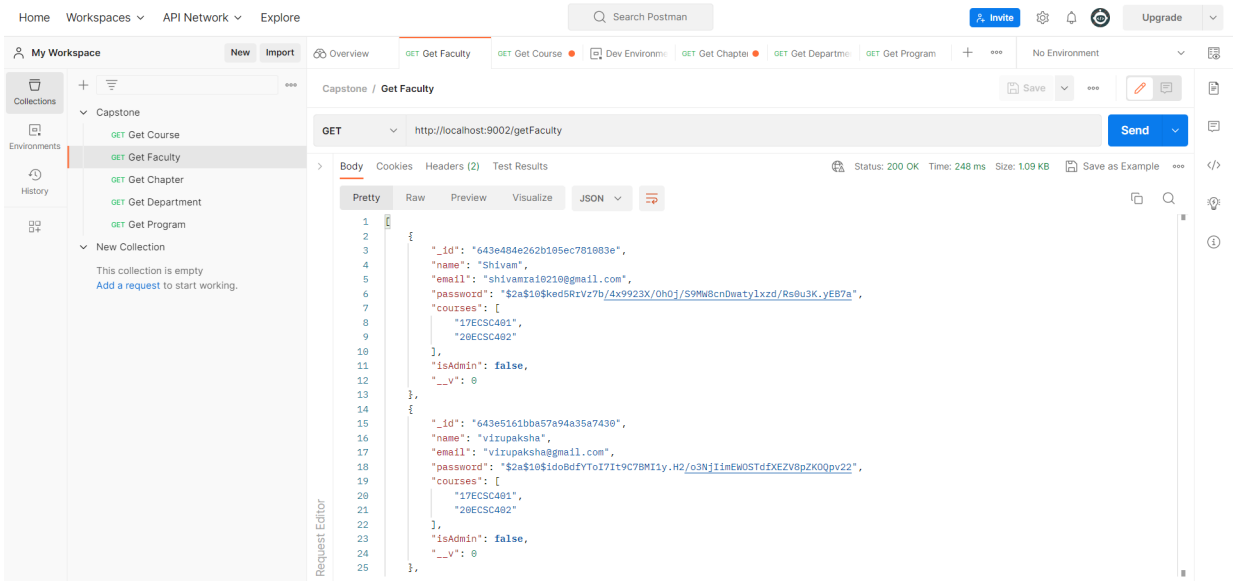


Figure 4.4: Get Faculty Details

A request to the server for retrieving the faculty details from the database. If the API is working fine, then it will send a response with the status code “OK” and will display all the retrieved faculty details.

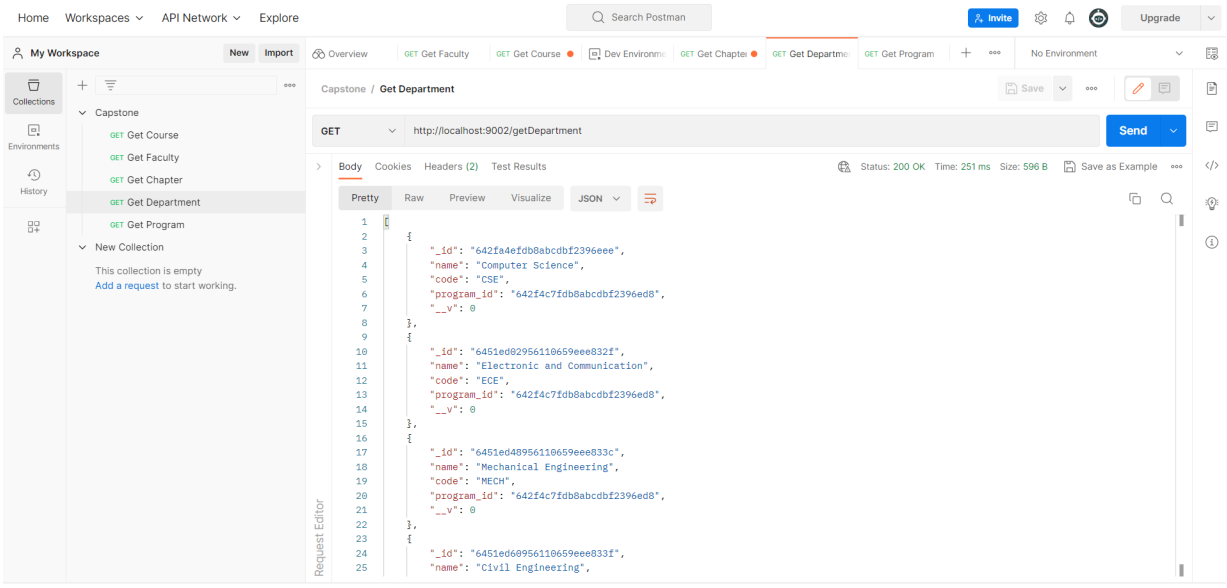


Figure 4.5: Get Department Details

A request to the server for retrieving the department details from the department table from the database. If the API is working fine, then it will send a response with the status code “OK” and will display all the retrieved department details.

Chapter 5

RESULTS AND DISCUSSION

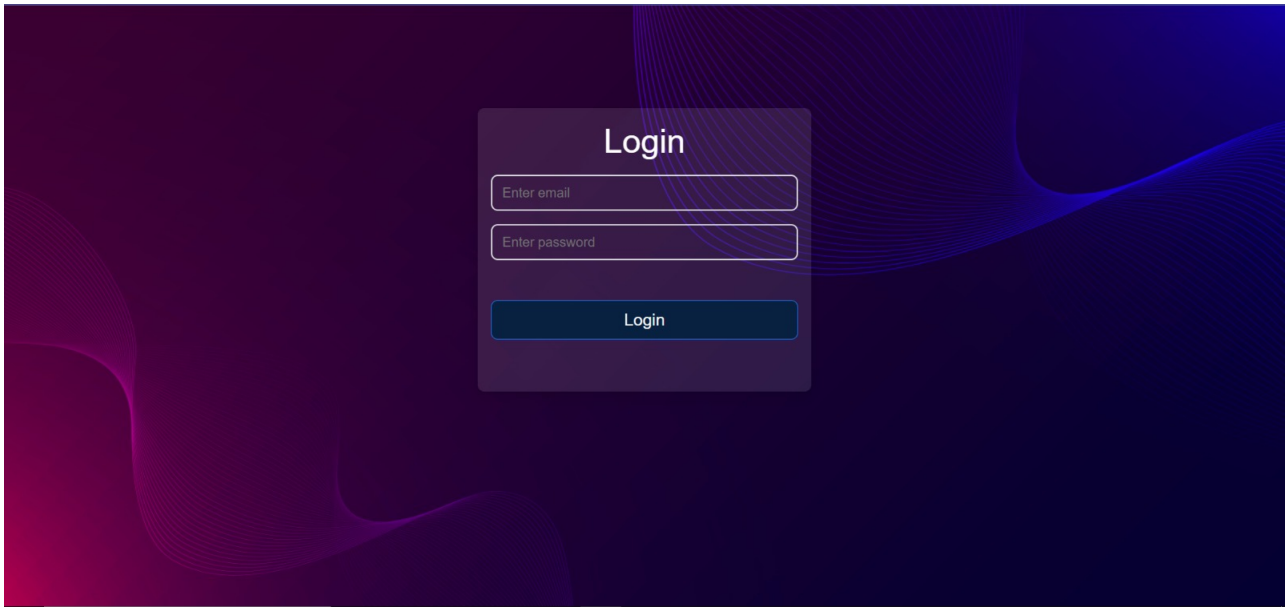


Figure 5.1: Login Page

This is the first page that gets rendered when a user tries to access the system. The login requires you to enter your email ID and password for the account. The account can be of faculty or admin which determines the accessibility to the system.

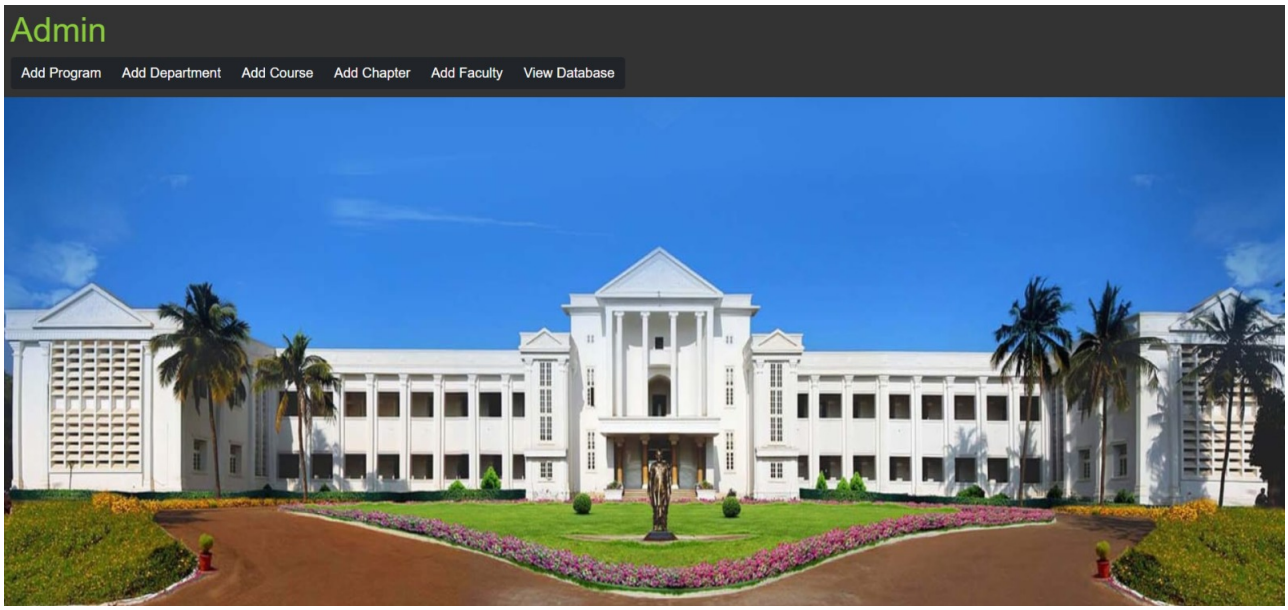


Figure 5.2: Admin Page

The admin page gets rendered upon logging in as an admin. It has a navigation menu for the admin to add programs, add departments, add courses, add faculty, and view the database. This set of actions can be performed only by an admin.

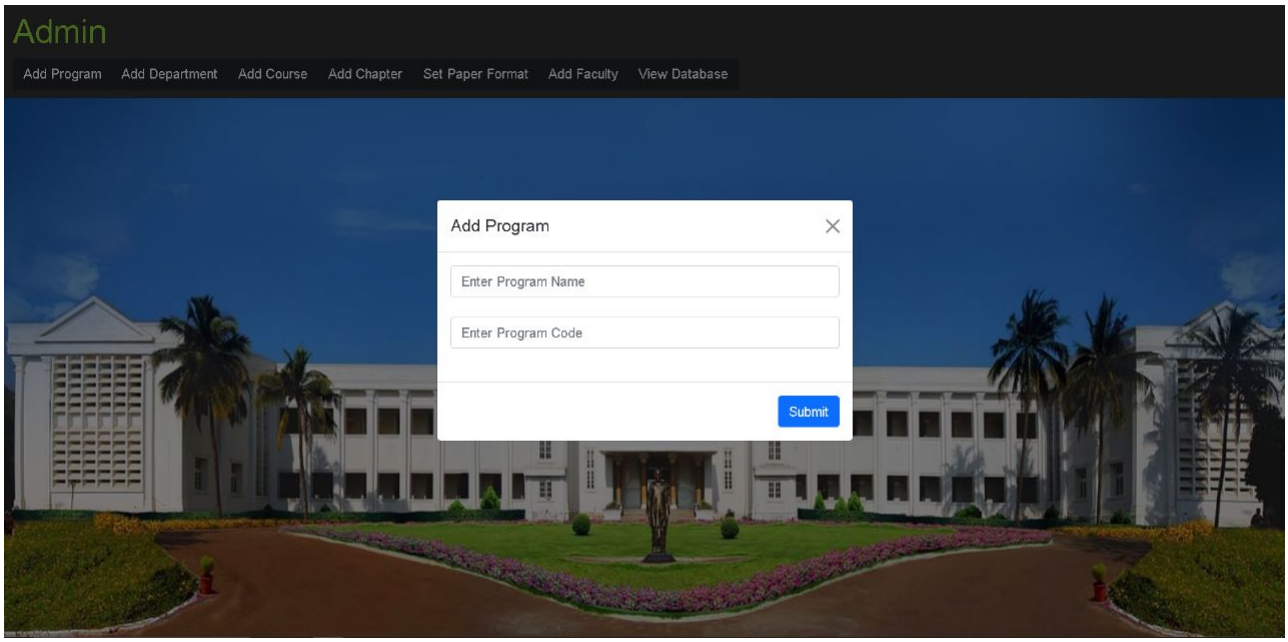


Figure 5.3: Add Program

This is the dialogue box that appears when the admin tries to add a new program. It requires the admin to enter the program name and program code. When the admin clicks on submit, a new program with the entered details gets added to the database. The database now gets updated upon the addition of new programs.

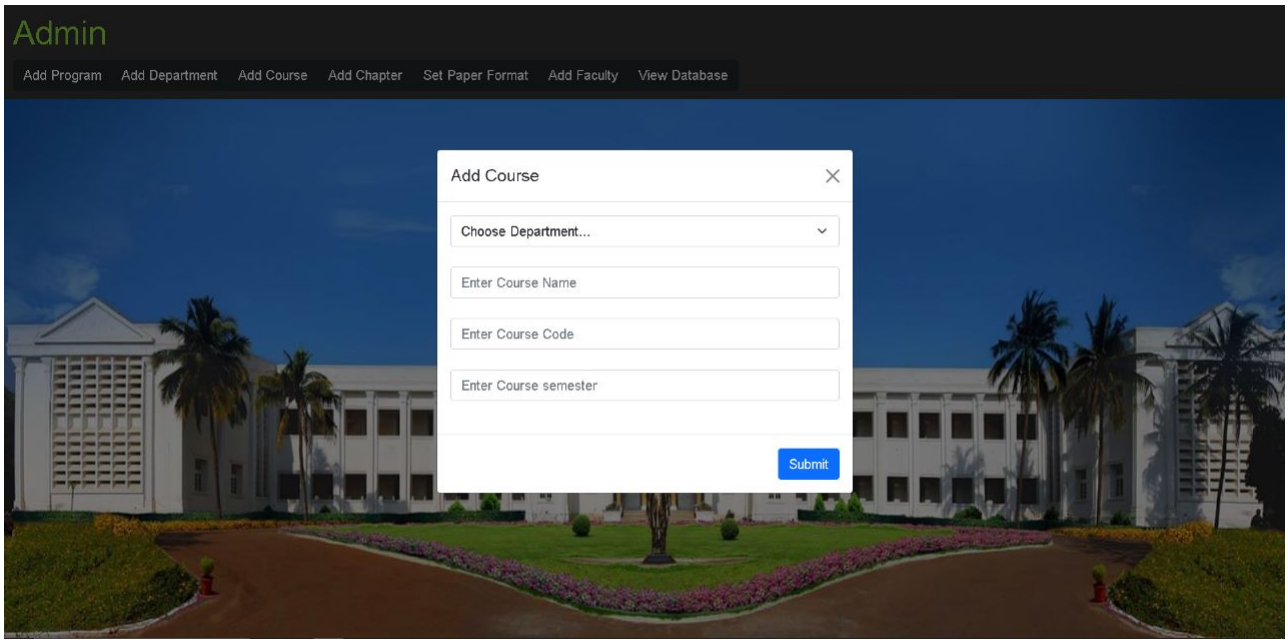


Figure 5.4: Add Course

This is the dialogue box that appears when the admin tries to add a new course. The input field choose department is a drop-down menu that shows the department that exists in the database of the collection department. The other fields are course name, code, and semester which are to be added compulsorily. After finishing the process of adding the data in the input fields, when the user clicks on submit, a new course with the entered details gets added to the database. The database now gets updated upon the addition of new courses.

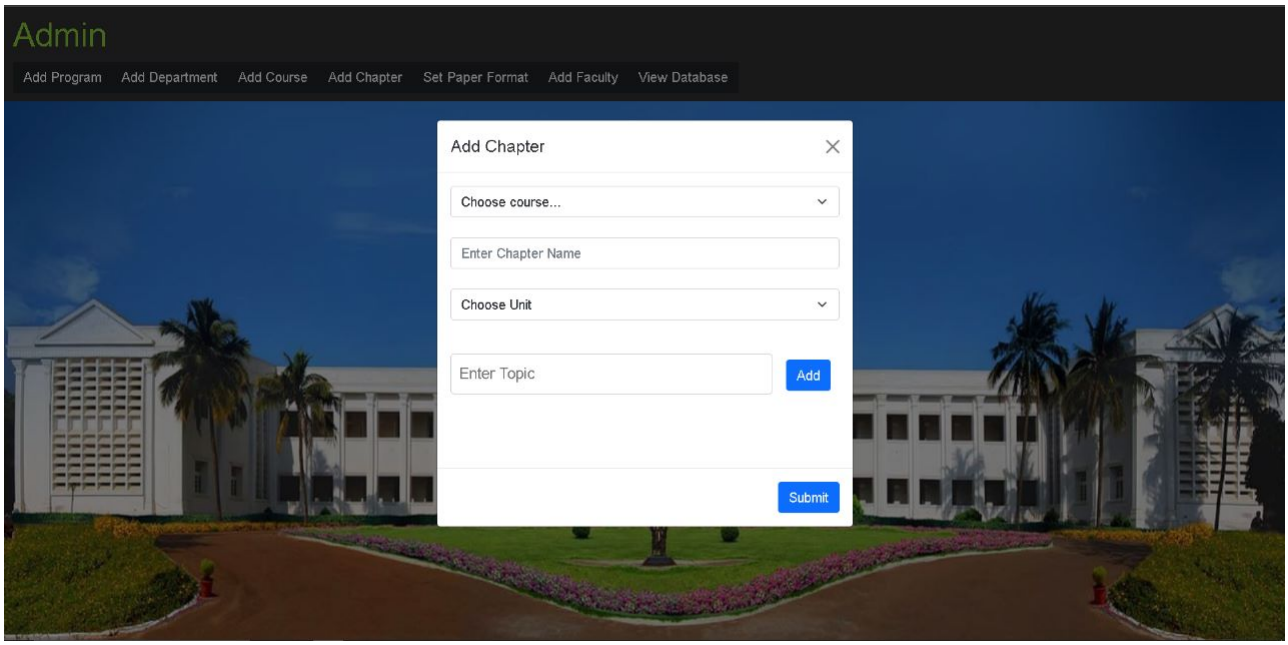


Figure 5.5: Add Chapter

This is the dialogue box that appears when the admin tries to add a new chapter. The input field Choose course is a drop-down menu that shows the course that exists in the database of the collection course. The chapter name needs to be entered manually by the admin. The input field Choose unit is again a drop-down menu that shows the units 1, 2, or 3. Enter topic is another field where more than one topic can be added serially. After finishing the process of adding the data in the input fields, when the user clicks on submit, a new chapter with the entered details gets added to the database. The database now gets updated upon the addition of new chapters.

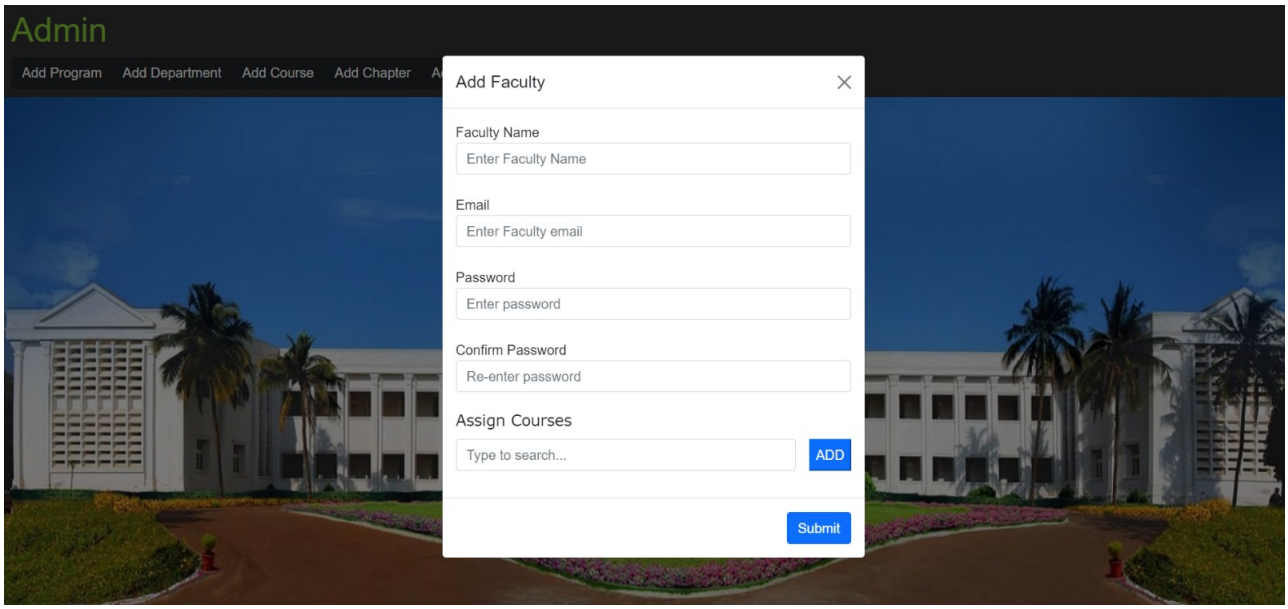


Figure 5.6: Add Faculty

This is the dialogue box that appears when the admin tries to add a new faculty. The admin is required to enter the faculty name and email. The password for the faculty access is to be added and is required to be confirmed by re-entering the same. The admin can assign many courses to the faculty one by one by clicking on add button. After finishing the process of adding the data in the input fields, when the user clicks on submit, a new faculty with the entered details gets added to the database. The database now gets updated upon the addition of new faculty.

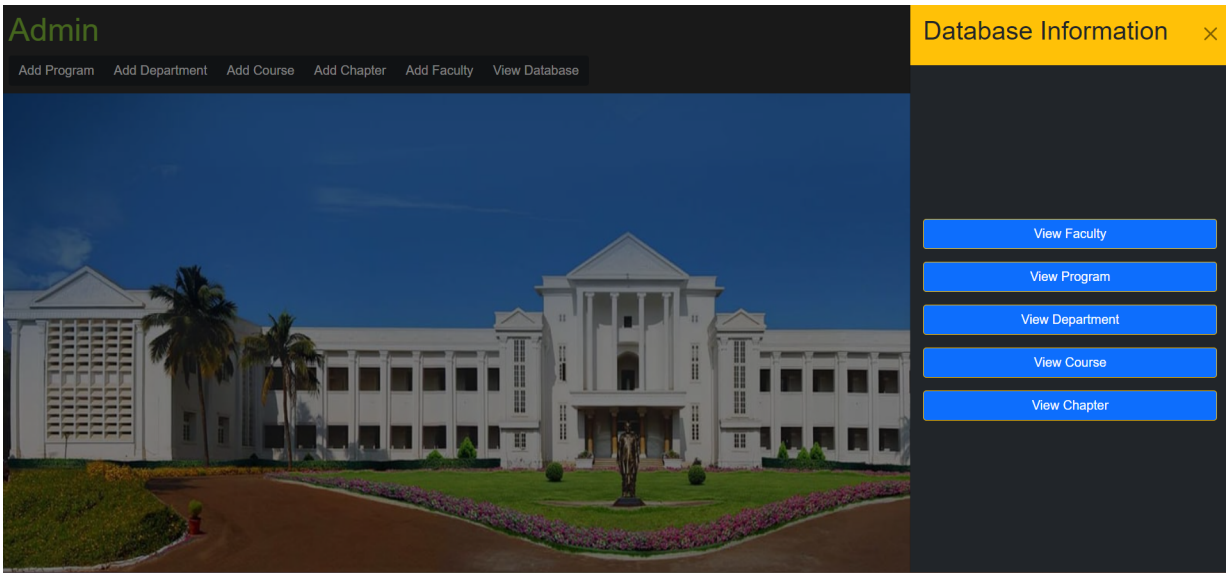
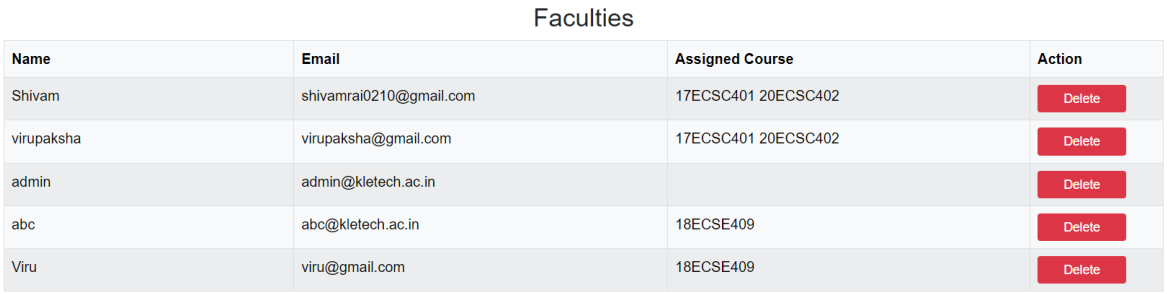


Figure 5.7: View Database

This is the layout that gets displayed when the user clicks on the view database. This is the database information that can be viewed by the admin. It consists of viewing the tables of faculty, program, department, course, chapter, semester course, and paper format.

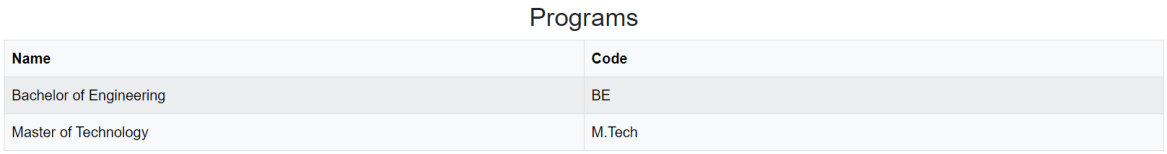


Faculties

Name	Email	Assigned Course	Action
Shivam	shivamrai0210@gmail.com	17ECSC401 20ECSC402	Delete
virupaksha	virupaksha@gmail.com	17ECSC401 20ECSC402	Delete
admin	admin@kletech.ac.in		Delete
abc	abc@kletech.ac.in	18ECSE409	Delete
Viru	viru@gmail.com	18ECSE409	Delete

Figure 5.8: View Faculty

This is the layout that gets displayed when the user clicks on view faculty. Here admin can view and delete the faculty data.



Programs

Name	Code
Bachelor of Engineering	BE
Master of Technology	M.Tech

Figure 5.9: View program

This is the layout that gets displayed when the user clicks on the view program. Here admin can view the program data.

Departments

Name	Code
Computer Science	CSE
Electronic and Communication	ECE
Mechanical Engineering	MECH
Civil Engineering	CVL

Figure 5.10: View Department

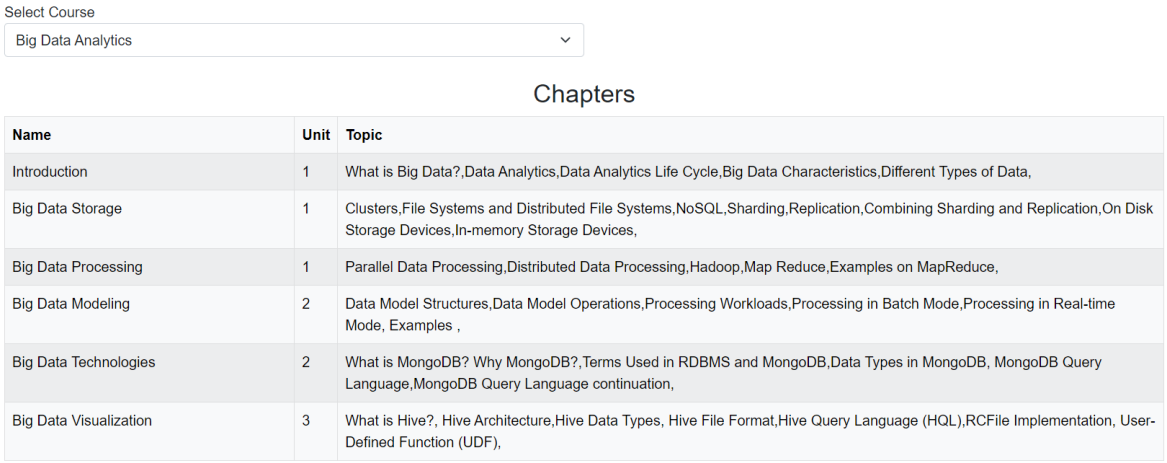
This is the layout that gets displayed when the user clicks on the view department. Here admin can view the department data.

Courses

Name	Course Code	Semester
Big Data Analytics	17ECSC401	7
Information Security	20ECSC402	7
Computer Organization and Architecture	20ECSC201	3
Software Engineering	15ECSC301	5
Cyber Security	19ECSE401	7
C# Programming and .NET	18ECSE409	8

Figure 5.11: View Courses

This is the layout that gets displayed when the user clicks on view courses. Here admin can view the course data.



Name	Unit	Topic
Introduction	1	What is Big Data?,Data Analytics,Data Analytics Life Cycle,Big Data Characteristics,Different Types of Data,
Big Data Storage	1	Clusters,File Systems and Distributed File Systems,NoSQL,Sharding,Replication,Combining Sharding and Replication,On Disk Storage Devices,In-memory Storage Devices,
Big Data Processing	1	Parallel Data Processing,Distributed Data Processing,Hadoop,Map Reduce,Examples on MapReduce,
Big Data Modeling	2	Data Model Structures,Data Model Operations,Processing Workloads,Processing in Batch Mode,Processing in Real-time Mode, Examples ,
Big Data Technologies	2	What is MongoDB? Why MongoDB?,Terms Used in RDBMS and MongoDB,Data Types in MongoDB, MongoDB Query Language,MongoDB Query Language continuation,
Big Data Visualization	3	What is Hive?, Hive Architecture,Hive Data Types, Hive File Format,Hive Query Language (HQL),RCFile Implementation, User-Defined Function (UDF),

Figure 5.12: View Chapters

This is the layout that gets displayed when the user clicks on view chapters. Here admin can view the chapter's data. admin can filter the courses which he wants then he can see chapters of respective courses.

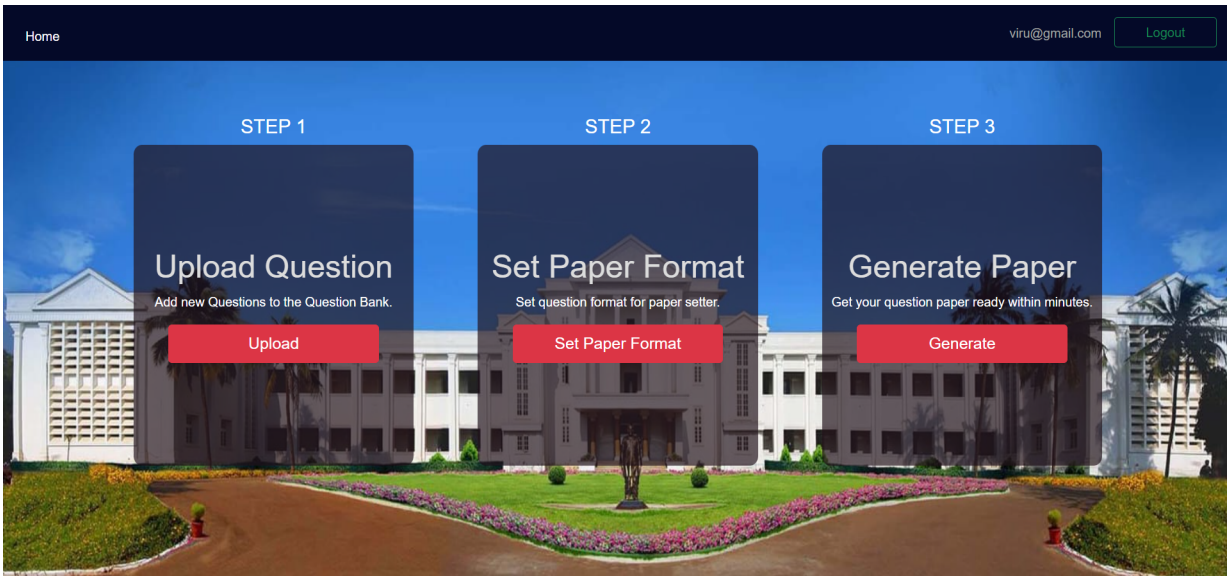


Figure 5.13: Home Page

This is the page that gets rendered as the dashboard when the user login is of a faculty. It has options to upload questions add new questions to the database and expand the question bank. The paper format can be set by the faculty. Another option is to generate a question paper.

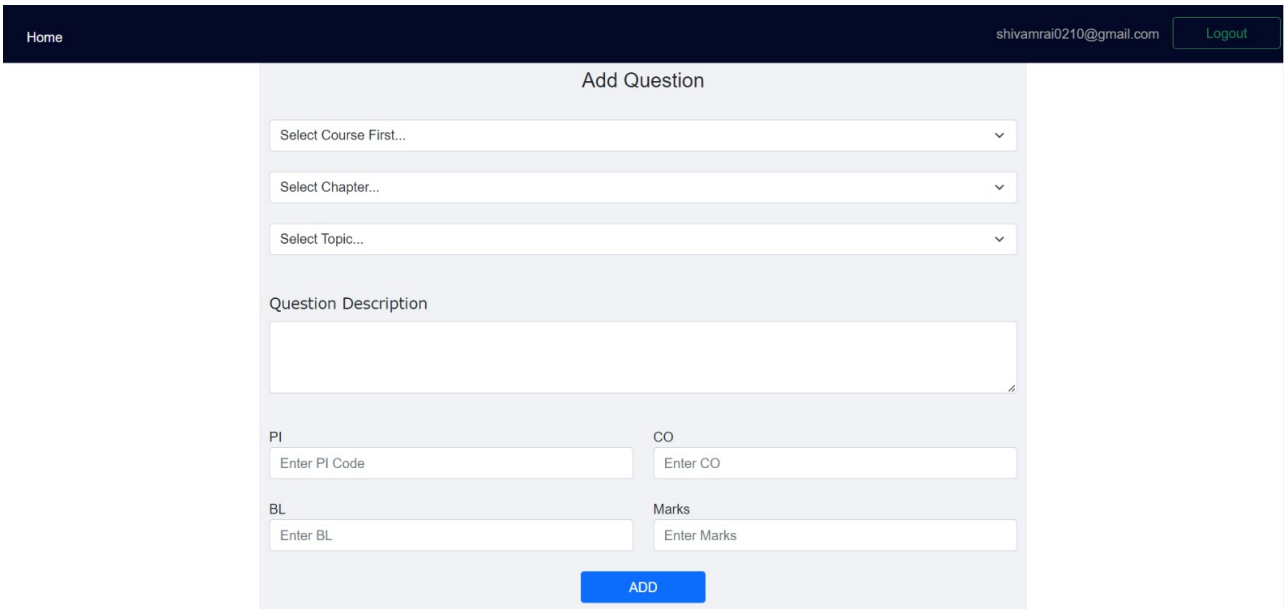


Figure 5.14: Upload Questions

This is the page that gets rendered when the faculty tries to upload a question. The faculty is required to enter the details related to a question as given in the form. The input field chooses course first is a drop-down menu that shows the course that exists in the database of the collection course. Similarly, for the other two fields, select the chapter and select a topic. Then the faculty is required to enter the question description in the text box. It is then mandatory to enter the PI Code, Course Outcome, Bloom's Level, and marks allotted for the question. After finishing the process of adding the data in the input fields, when the user clicks on add, a new question with the entered details gets added to the database. The database now gets updated upon the addition of new questions in the question bank.

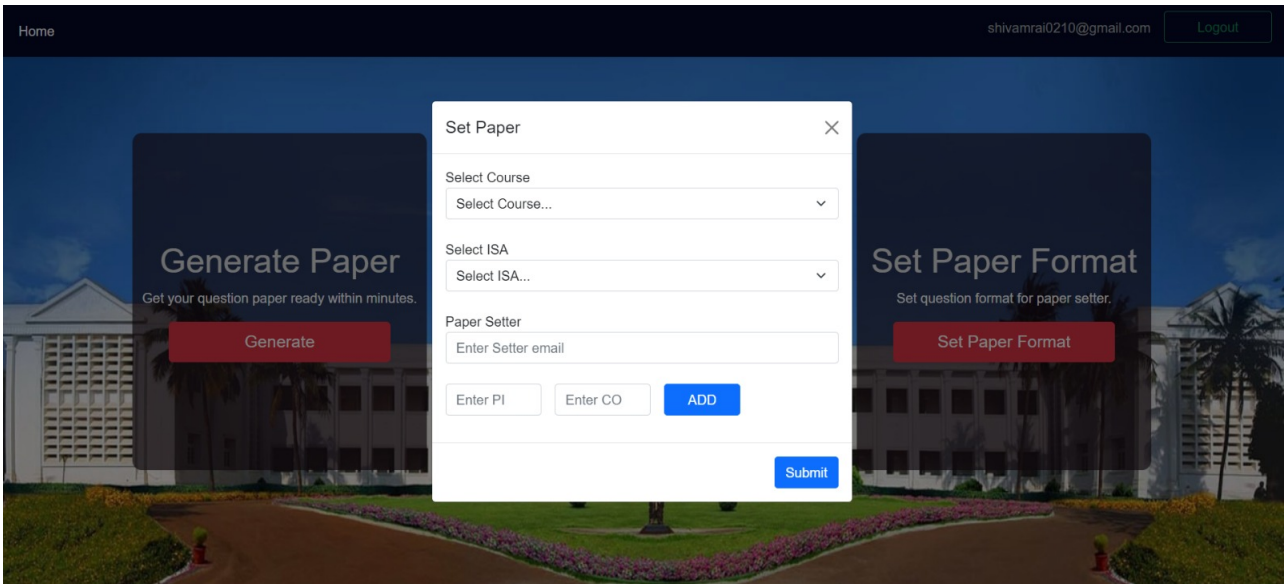


Figure 5.15: Set Paper

This is the dialogue box that appears when the faculty tries to add a new faculty. The input field chooses course first is a drop-down menu that shows the course that exists in the database of the collection course. The input field selects ISA is also a drop-down that contains options for ISA 1 or 2. The email of the paper setter is to be mentioned along with entering a set of the PI and CO codes. These PI and CO codes can be more than one which can be added by clicking on the button serially. After finishing the process of adding the data in the input fields, when the user clicks on submit, a new paper format with the entered details gets added to the database.

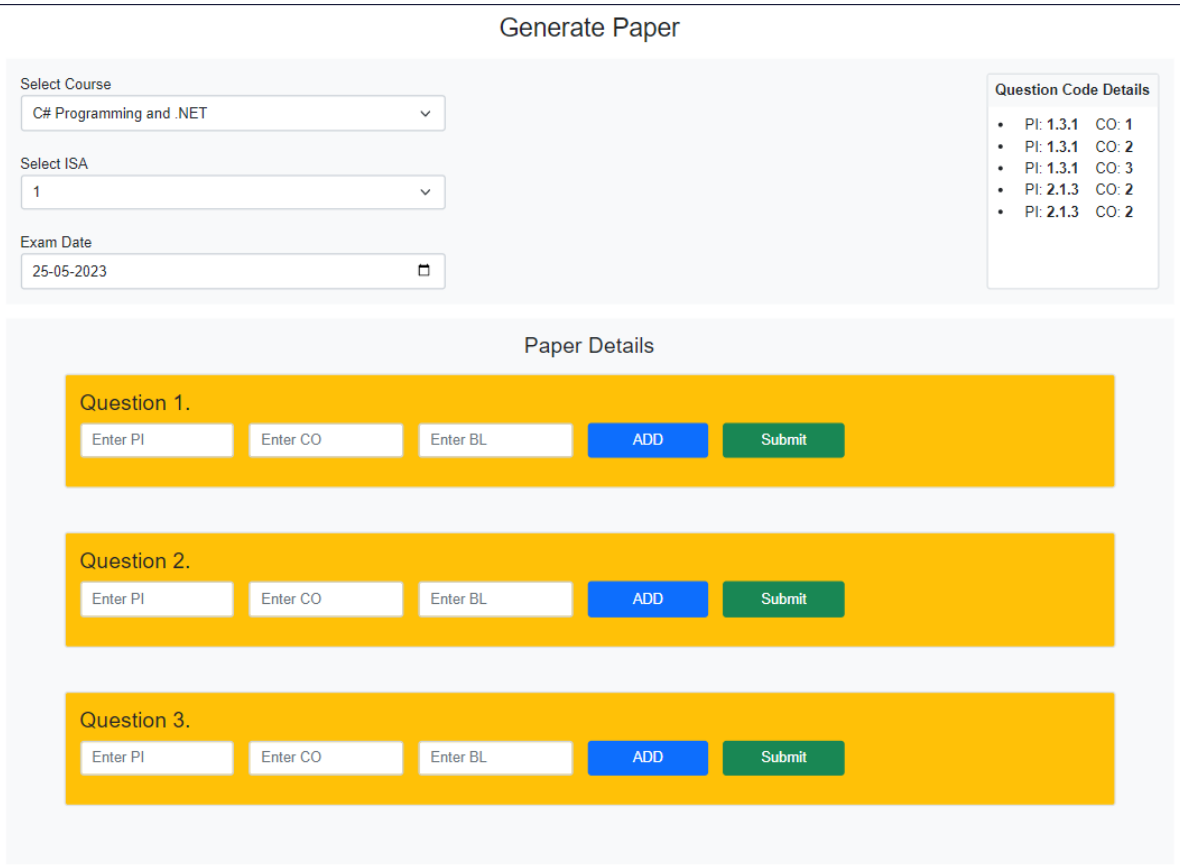


Figure 5.16: Paper Details

This is the page that gets rendered when the faculty tries to generate a question paper. The faculty is required to select a course name from the drop-down menu provided for selecting a course. After selecting the g course name, the question paper details appear on the right with the available PI and CO codes format for setting the question paper. The faculty can then select the ISA from the drop-down menu which has options of ISA 1 or 2. The exam date can be entered from the calendar infielder. After this, the paper details appear.

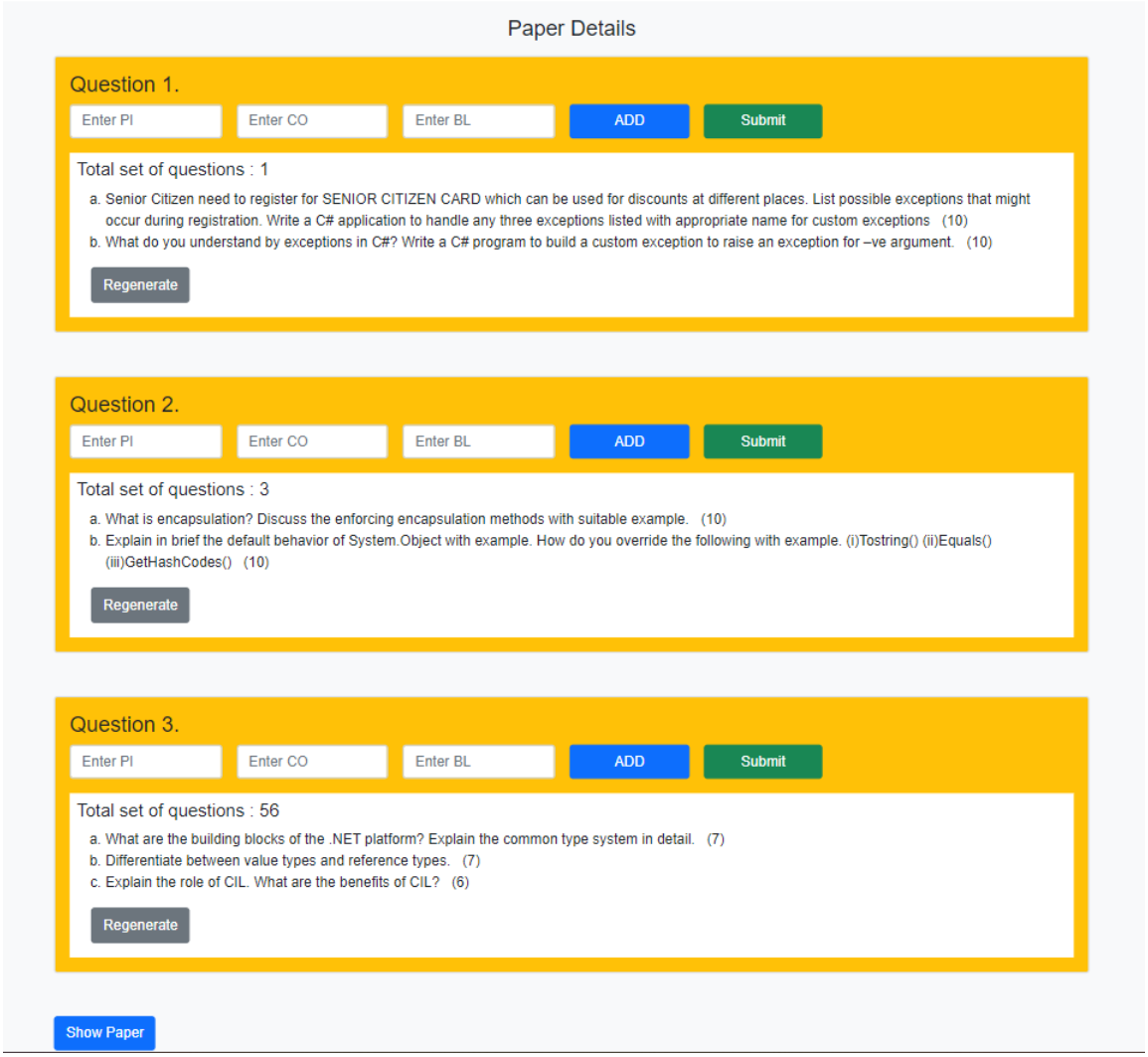
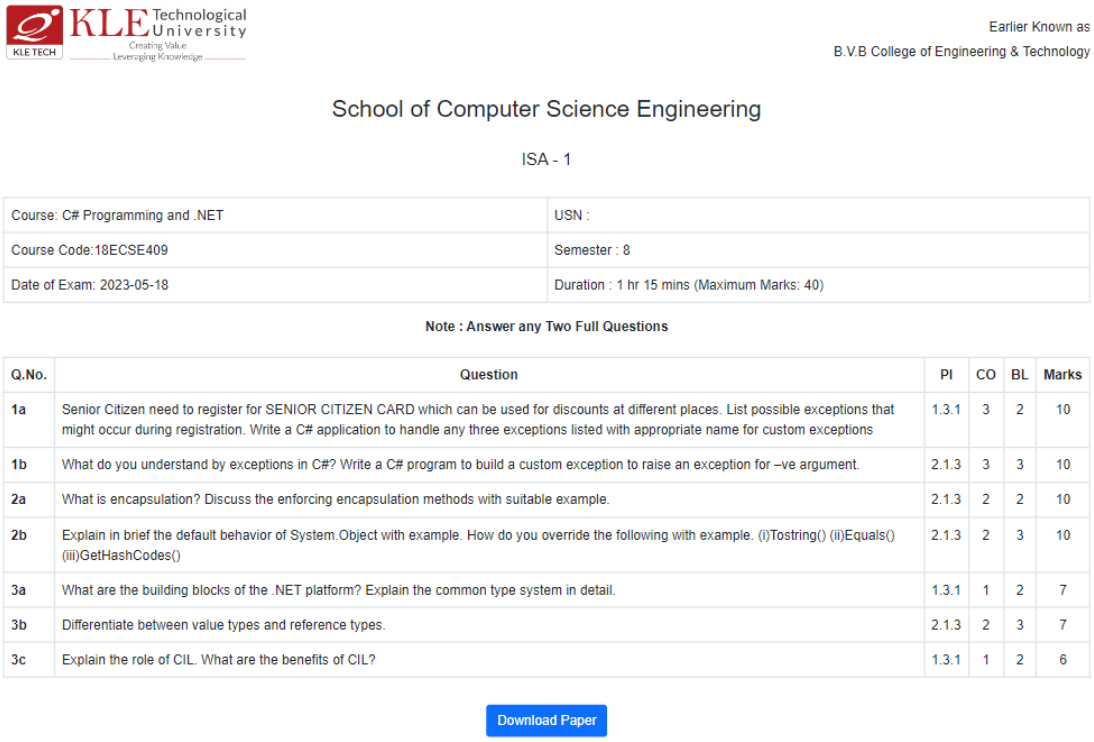


Figure 5.17: Generate Paper

This is the page that gets rendered when the faculty tries to generate a question paper. The faculty can add a set of PIs and COs for each question and finally click on submit after completing the process of adding the PIs and COs. If the set of questions is not satisfied then they can click on the Regenerate button. It will regenerate the new set of questions.



KLE Technological University
Creating Value, Leveraging Knowledge
KLE TECH

Earlier Known as
B.V.B College of Engineering & Technology

School of Computer Science Engineering

ISA - 1

Course: C# Programming and .NET	USN :
Course Code:18ECSE409	Semester : 8
Date of Exam: 2023-05-18	Duration : 1 hr 15 mins (Maximum Marks: 40)

Note : Answer any Two Full Questions

Q.No.	Question	PI	CO	BL	Marks
1a	Senior Citizen need to register for SENIOR CITIZEN CARD which can be used for discounts at different places. List possible exceptions that might occur during registration. Write a C# application to handle any three exceptions listed with appropriate name for custom exceptions	1.3.1	3	2	10
1b	What do you understand by exceptions in C#? Write a C# program to build a custom exception to raise an exception for –ve argument.	2.1.3	3	3	10
2a	What is encapsulation? Discuss the enforcing encapsulation methods with suitable example.	2.1.3	2	2	10
2b	Explain in brief the default behavior of System.Object with example. How do you override the following with example. (i)Tostring() (ii)Equals() (iii)GetHashCodes()	2.1.3	2	3	10
3a	What are the building blocks of the .NET platform? Explain the common type system in detail.	1.3.1	1	2	7
3b	Differentiate between value types and reference types.	2.1.3	2	3	7
3c	Explain the role of CIL. What are the benefits of CIL?	1.3.1	1	2	6

Download Paper

Figure 5.18: Show Paper

This is the page that gets rendered when the faculty tries to see a generated question paper. when the faculty click on the show paper button it will show the generated question paper. After that, the faculty can download the paper in pdf format.

KLE Technological University
Creating Value
Leveraging Knowledge

Earlier Known as
B.V.B College of Engineering & Technology

School of Computer Science Engineering

ISA - 1

Course: C# Programming and .NET	USN :
Course Code:18ECSE409	Semester : 8
Date of Exam: 2023-05-15	Duration : 1 hr 15 mins (Maximum Marks: 40)

Note : Answer any Two Full Questions

Q.No.	Question	PI	CO	BL	Marks
1a	What are the building blocks of the .NET platform? Explain the common type system in detail.	1.3.1	1	2	7
1b	Differentiate between value types and reference types.	2.1.3	2	3	7
1c	Mention some of exceptions defined in System.SystemException and System.SystemApplication.	2.1.3	3	2	6
2a	Explain the role of JIT compiler. Explain boxing and unboxing with example.	2.1.3	2	2	6
2b	What do you understand by exceptions in C#? Write a C# program to build a custom exception to raise an exception for –ve argument.	2.1.3	3	3	10
2c	How do you configure a C# project in VS .NET IDE?	1.3.1	1	2	4
3a	Explain with a neat diagram how generation 0 objects are promoted to generation 1.	2.1.3	3	2	6
3b	What are the three pillars of object-oriented programming in C#? Differentiate between “is a” & “has a” relationships. What is the	2.1.3	2	3	8

Q.No.	Question	PI	CO	BL	Marks
	difference between enforcing encapsulation using the traditional accesor and mutators as well as class properties?				
3c	How has the life of programmer changed with the evolution of the C# language?	1.3.1	1	2	6

Chapter 6

CONCLUSION

The challenge that is faced by majority of the course teachers in universities has been solved to an extent. We have developed an automated process of question paper generation. This is fast, streamlined, randomized and secure. All the action executed tasks by the system are automated which optimizes the storage, space, bias and security. The initial steps like registering faculty, authenticating user, admin panel have all been implemented. The University database for storing faculty details, course details and program details have successfully been implemented. The algorithm for question paper is generation is implemented and the question papers are generated based on the OBE values provided in the prompt. The question bank can be expanded and it automatically generates a variety of sets of questions.

Chapter 7

REFERENCES

1 Dubey Harish, Tamore Hardik, Padhi Sagar, Prof. Manisha Bharambe, "Automatic Question Paper Generator System", INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY (IRJET), MAY 2020.

2 Gajalakashmi D, Johnson Jero S, "Automatic Question Paper Generator Using Bloom's Taxonomy", Journal of Computing and Intelligent Systems, Nov 2020.

3 Parth Shah, Uzair Faquih, Rutuja Devkar, Yogesh Shahare, "An Intelligent Question Paper Generator using Randomized Algorithm", INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH and TECHNOLOGY (IJERT), JAN 2022.

4 Shivam Kr. Dwivedi, Shekhar S. Rawat, Sahil Khanna, Saurabh Rathour, Krishan Kumar, "QBase – A Bloom Taxonomy based Question Paper Generator", INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY (IRJET), MAY 2020.

5 Dwivedi, Roshni, Kriti Kashyap, Pranita Dhoke, Swati Korhale, and G. S. Navale. "Adaptive Question Paper Generation System." (2017).

6 N. H. I. Teo, N. A. Bakar and S. Karim, "Designing GA-Based Auto-Generator of Examination Questions," 2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation, Malta, Malta, 2012, pp. 60-64, doi: 10.1109/EMS.2012.69.

7 Vijay KrishanPurohit', Abhijeet Kumar', Asma Jabeen, Saurabh Srivastava, R H Goudar, Shivanagowda, "Design of Adaptive Question Bank Development and Management System", 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012.

8 Suraj Kamya, Madhuri Sachdeva, Navdeep Dhaliwal and Sonit Singh, "Fuzzy Logic Based Intelligent Question Paper Generator" IEEE International Advance Computing Conference (IACC), 2014.

9 Rahmalan, H., Abidin, Z.Z., Abas, Z.A., Rahman, A.A., Shibghatullah, A.S. and Rhui-Yong, O., 2015, December. Prototype deployment of examination and quiz system

(EQs) with outcome based education in networking system. In 2015 IEEE International Conference on Communication, Networks and Satellite (COMNESTAT) (pp. 43-47). IEEE.

10 M. Fatangare, R. Pangare, S. Dorle, U. Biradar and K. Kale, "Android based exam paper generator (Android based E-PAGE)," 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2018, pp. 881-884, doi: 10.1109/ICISC.2018.8398926.

11 P. Pisat, D. Modi, S. Rewagad, G. Sawant and D. Chaturvedi, "Question paper generator and answer verifier," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017, pp. 1074-1077, doi: 10.1109/ICECDS.2017.8389603.

12 G. Nalawade and R. Ramesh, "Automatic Generation of Question Paper from User Entered Specifications Using a Semantically Tagged Question Repository," 2016 IEEE Eighth International Conference on Technology for Education (T4E), Mumbai, India, 2016, pp. 148-151, doi: 10.1109/T4E.2016.038.

13 Noor Hasimah Ibrahim Teo, Nordin Abu Bakar and Moamed RezduanAbd Rashid, “Representing Examination Question Knowledge into Genetic Algorithm”, IEEE Global Engineering Education Conference (EDUCON), 2014.

14 Mojitha Mohandas , Aishwarya Chavan , Rasika Manjarekar and Divya Karekar, “Automated Question Paper Generator System”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 12, December 2015

15 Gauri Nalawade and Rekha Ramesh, “Automatic Generation of Question Paper from User Entered Specifications using a Semantically Tagged Question Repository”, 2016 IEEE 8th International Conference on Technology for Education

16 Prateek Pisat, Shrimangal Rewagad, Devansh Modi and Ganesh Sawan, “Question Paper Generator and Answer Verifier”, International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017)

17 Tejas Barot and Poornima Salunke “Automatic Question Paper Generator System”, International Journal of Scientific Research Engineering and Technology (IJSRET), Volume 6, Issue 4 , April 2017

18 Amit Sanjay Khairnar, Bhagwat Chintaman Jadhav, Rahul Birhade, Pramod Patil, “Automatic Question Paper Generator”, International Journal For Technological Research In Engineering Volume 4, Issue 9, May-2017

19 Prof. Mrunal Fatangare, Rushikesh Pangare, Shreyas Dorle, Uday Biradar, Kaustubh Kale,"Android Based Exam Paper Generator", Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018)

20 Vafeiadou, Catherine, PantelisVasiloudis, and Minas Dasygenis. "Online automatic examination system for digital circuits." Modern Circuits and Systems Technologies (MOCAST), 2016 5th International Conference on.IEEE, 2016.

21 Banerjee, Shilpi, and ChandrashekarRamanathan. "Automatic Generation of Valid and Equivalent Assessment Instruments." Emerging Issues in Smart Learning.Springer Berlin Heidelberg, 2015.97-104.

22 Radha, D., et al. "A Novel Approach to Assist Faculties In Question Paper Generation." International Journal of Computer Applications 150.10 (2016).

23 International Journal of Scientific and Research Publications, Volume 5, Issue 1, January 2015 1 ISSN 2250-3153.

24 Kapil Naik, Shreyas Sule, Shruti Jadhav, and Surya Pandey, "Automatic Question Paper Generation using Randomization Algorithm," International Journal of Engineering and Technical Research, vol. 2, issue 12, December 2014.

25 Automatic Cloze-Questions Generation by Annamaneni Narendra, Manish Agarwal, and Rakshit shah LTRC, IIIT-Hyderabad, India.

26 Automatic Question Generation using Discourse Cues by Manish Agarwal, Rakshit Shah, and Prashanth Mannem.

27 M. Kaliannan and S. D. Chandran, "Empowering students through outcome-based education (OBE)", 2006.

28 "Education and skills for inclusive and sustainable development beyond 2015 Thematic Think Piece", 2012.

29 S. Gombe, T. Suandi, I. A. Ismail and Z. Omar, "Extension Worker Competencies Needed for Effective Management of Self-Help Groups (SHG) in Gombe State", Int. J. Educ. Train., vol. 1, no. 1, pp. 1-7, 2015.

30 R. Bridgstock, B. Goldsmith, J. Rodgers and G. Hearn, "Creative graduate pathways within and beyond the creative industries", J. Educ. Work, vol. 28, no. 4, pp. 333-345, 2015.

31 Mohd M. N. Arshad and Ab Z. Malik, "QUALITY OF HUMAN CAPITAL AND LABOR PRODUCTIVITY: A CASE OF MALAYSIA", Int. J. Econ. Manag. Account., vol. 23, no. 1, pp. 37-55, 2015.

32 M. S. Ariff, N. Zaidin and N. Sulong, "Total Quality Management Implementation in Higher Education; Concerns and Challenges Faced by the Faculty", 12 the International Conference on ISO, vol. 9000, pp. 1-23, 2007.

33 K. K. Ali, R. Salleh and M. Sabdin, "A Study on the Level of Ethics at a Malaysian Private Higher Learning Institution: Comparison between Foundation and Undergraduate Technical-based Students", Int. J. Basic Appl. Sci., vol. 10, no. 05, pp. 20-30, 2010.

34 F. Bouslama, A. Lansari, A. Al-Rawi and A. A. Abonamah, "A Novel outcome-based educational model and its effect on student learning curriculum development and assessment", J. Inf. Technol. Educ., vol. 2, pp. 203-214, 2003.

35 M. R. Mamat, M. S. Rasul and A. Mustapha, "Outcome-based education implementation in malaysian polytechnic", Int. J. Educ. Res., vol. 2, no. 11, pp. 437-450, 2014.

36 M. T. Telsang, "Outcome Based Education - Design Delivery and Assessment of Product Design and Development Course at Undergraduate Engineering Program", J. Eng. Educ. Transform., pp. 145-149, 2015.

37 S. Jimaa, "The impact of assessment on students learning", Procedia - Soc. Behav. Sci., vol. 28, pp. 718-721, 2011.

38 P. Black and D. Wiliam, "Inside the Black Box: Raising Standards Through Classroom Assessment", Phi Delta Kappan, vol. 80, no. 2, pp. 139-148, 1998.

39 S. Mohamad, Z. Tukiran, R. M. Hanifa, A. Ahmad and M. Som, "An Evaluation of Assessment Tools in Outcome-based Education: A Way Forward", vol. 3, no. 11, pp. 336-343, 2012.

40 W. Ibrahim, Y. Atif, K. Shuaib and D. Sampson, "A Web-Based Course Assessment Tool with Direct Mapping to Student Outcomes", Educ. Technol. Soc., vol. 18, no. 2, pp. 46-59, 2015.

41 L.-H. Hsia, I. Huang and G.-J. Hwang, "A web-based peer-assessment approach to improving junior high schools students' performance self-efficacy and motivation in performing arts courses", Br. J. Educ. Technol., pp. 1-15, 2015.

42 Y. Z. Y. Zhenming, Z. L. Z. Liang and Z. G. Z. Guohua, "A novel web-based online examination system for computer science education", 33rd Annu. Front. Educ. 2003. FIE 2003, vol. 3, no. 222, pp. 7-10, 2003.

43 Y. Li and J.-Q. Tang, "The Research of Examination Paper Generation Based on Index System Metrics and Multi-Objective Strategy", J. Softw. Eng. Appl., vol. 05, no. 08, pp. 634-638, 2012.

44 Haris, S. S., and Omar, N. (2012). A rule-based approach in Bloom's Taxonomy question classification through natural language processing. In 2012 7th International Conference on Computing and Convergence Technology (ICCCT) (pp. 410–414).

45 Gangar, F. K., Gori, H. G., and Dalvi, A. (2017). Automatic question paper generator system. International Journal of Computer Applications, 166, 42–47.

46 Nalawade, G., and Ramesh, R. (2016). Automatic generation of question paper from user entered specifications using a semantically tagged question repository. In 2016 IEEE Eighth International Conference on Technology for Education (T4E) (pp. 148–151), Mumbai.

47 Ilango Sivaraman, S., and Krishna, D. (2015). Bloom's taxonomy—application in exam papers assessment. International Journal Of Multidisciplinary Sciences and Engineering, 6(9).

48 von Konsky, B., Zheng, L., Parkin, E., Huband, S., and Gibson, D. (2018). Parts of speech in bloom's taxonomy classification.

49 Kanakaraddi, S. G., and Nandyal, S. S. (2018). Survey on parts of speech tagger techniques. In 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT) (pp. 1–6).

50 Tian, Y., and Lo, D. (2015). A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports. In 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER).

51 Sutskever, I., Martens, J., and Hinton, G. E. (2011). Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11) (pp. 1017–1024).

52 Wang, Y. (2017). A new concept using LSTM neural networks for dynamic system identification. In 2017 American Control Conference (ACC) (pp. 5324–5329).

53 Yang, S. (2020). LSTM and GRU neural network performance comparison study. In International workshop on electronic communication and artificial ıntelligence (IWECAI).

54 Mohammed, M., and Omar, N. (2020). Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec. PLosONE, 15(3):e0230442.

55 Wang, J., Liu, T., Luo, X., and Wang, L. (2018). An LSTM approach to short text sentiment classification with word embeddings. ROCLING.

OBE Based Model Question Paper Generator

ORIGINALITY REPORT

15%	15%	4%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijert.org Internet Source	5%
2	Saroja V Siddamal, Suhas B Shirol, Shraddha Hiremath, Nalini C Iyer. "Towards Sustainable Integrated Model for Skill Development: A Collaborative Approach", Procedia Computer Science, 2020 Publication	4%
3	origin.geeksforgeeks.org Internet Source	2%
4	www.w3cschoool.com Internet Source	1%
5	sahil-more.medium.com Internet Source	1%
6	www.javatpoint.com Internet Source	1%
7	glisc.info Internet Source	<1%
8	yx.co-aol.com Internet Source	

		<1%
9	utpedia.utp.edu.my Internet Source	<1%
10	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
11	Beginning T-SQL 2012, 2012. Publication	<1%
12	elibrary.buse.ac.zw:8080 Internet Source	<1%
13	publications.polymtl.ca Internet Source	<1%
14	www.slideshare.net Internet Source	<1%
15	"Information and Communication Technology for Intelligent Systems", Springer Science and Business Media LLC, 2019 Publication	<1%
16	Yuan-Bin, Chen, and Dai Jie. "Design on Algorithm of Automatic Test Papers Generation for Examination System of Electric Energy Measurement", 2012 International Conference on Computer Science and Service System, 2012. Publication	<1%