PROJECT REPORT

On

“Smart Question Bank”

ABSTRACT

This project is the smart approach and solution of subject-wise preparation with different types of questions, like descriptive-type questions, multiple types of questions, etc., on a web platform where teachers will give the questions. The project aims to create a database of questions on an IT-supported platform and provide the faculty with options to create different types of question papers of varying difficulty levels on demand. The project focuses on the development and deployment of a smart solution for subject-wise preparation of question banks for implementation at an institution. The solution is expected to be used primarily by members of the faculty who would create questions pertaining to different units of different subjects of varying levels of difficulty.

The planned solution is expected to not only maintain a repository for such questions but also provide the stakeholders with the option to create question papers speedily and on demand. It will be very helpful for the online assessment. The application will have 3 types of users: teachers, reviewers, and admins. The technologies used in this project include ReactJS, Material UI, NodeJS, Express.js, and MariaDB for the backend database, React-Redux, and MVC Framework. Two different types of Dashboards will open for the teacher’s login and the reviewer's login on the other side. This web page is to create questions for students and check how much they are preparing.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

1.1 Background

The primary focus of this Smart Question Bank project is an online platform where teachers can upload different types of questions (MCQ, long answer type, etc.). This project aims to create a database of questions on the IT-supported platform and provide the faculty with options to create different types of question papers of varying difficulty levels on demand.

As the prerequisites of this project, we need knowledge of web development with HTML, CSS, JavaScript, ReactJS, Material UI, NodeJS, Express.js, and MariaDB. We need good and proper implementation planning for the whole project. We have to create a Web application using these technologies. As the outcome of this project, the teachers will be able to use it as per their requirements.

1.2 Purpose of the Project

The purpose of the project is to develop a web application that is a smart approach to subject-wise question bank preparation for the faculty. It will be helpful for the students to practice questions with different subjects. Also, teachers can create and make different sections according to the difficulty level of the question. We can give custom categorization, customization, and a specific review system in this application. This is the business perspective of this project.

Who will benefit from it?

- Faculties can refer to the Question bank for the purpose of setting the question papers for internal assessment as well as for their term-end examination.
- Teachers can also refer to the question bank for continuous evaluation of students. After completing one unit of any course, teachers can give some assignments to students to do based on the question bank

1.3 Problem Statement

Nowadays, a huge part of the studying system and preparation for exams has become online-based based to the post-COVID situation. Teachers do not get the students for face-to-face checking their preparation face-to-face every time. Students have to search for different tutorial-based and quiz-based sites to solve the questions. For their preparation and knowledge testing, they have searched for subject-wise websites and questions, which sometimes became time-consuming before their busy schedule of exams. It will be very helpful and efficient learning for them if they can get all the subject-wise questions on one platform for the best results. Based on this problem scenario, we have launched this project theme “Smart Question Bank”, where teachers can create several questions for all the subjects of their course. The existing web application in this area has not been cost-optimized; our approach is to build it efficiently.

1.4 Objective

The objective is to develop and launch a smart solution of subject-wise preparation with MCQs, where the faculty will be able to create the questions from their end to check the students. This platform will encourage students to study and prepare efficiently for the exams. Faculties can refer to the Question bank for the purpose of setting the question papers for internal assessment as well as for their term-end examination. Teachers can also refer to the question bank for continuous evaluation of students. After completing one unit of any course, teachers can give some assignments to students to do by using this question bank. This ‘Smart Question bank’ aims to give the students surprise questions and to judge them properly on how prepared they are.

1.5 Structure of Project

1.5.1 Outline

The entire project scope has been divided into 3 primary phases:

Phase 1: Question creation & audit

- Sign-up or login (authentication)
- Support the creation of the following question pattern:
    - MCQ
    - General question
- Map questions to the following type:
    - Objective
    - Definition/Naming
    - Short Questions
    - Explanation-Based Questions
    - Questions on Reasoning (If applicable)
    - Application-Based Questions
    - Short Notes
- Support questions for the following difficulty level:
    - Easy
    - Medium
    - Hard
- Support questions to be tagged to multiple papers/subjects belonging to different academic programs.
- Support the following user roles:
    - Contributor
    - Reviewer
    - Administrator

- Peer & expert review of the questions submitted by other faculty members of the concerned academic department.
- Basic checks & validations

Phase 2: Creation of Question Papers

- Efficient UI mechanism for developing question papers for defined marking templates
- Save & export the question papers

Phase 3: Enhancements

- UI enhancements
- Dashboard functionality
- Additional checks & validations.

1.5.2 Project Flow:

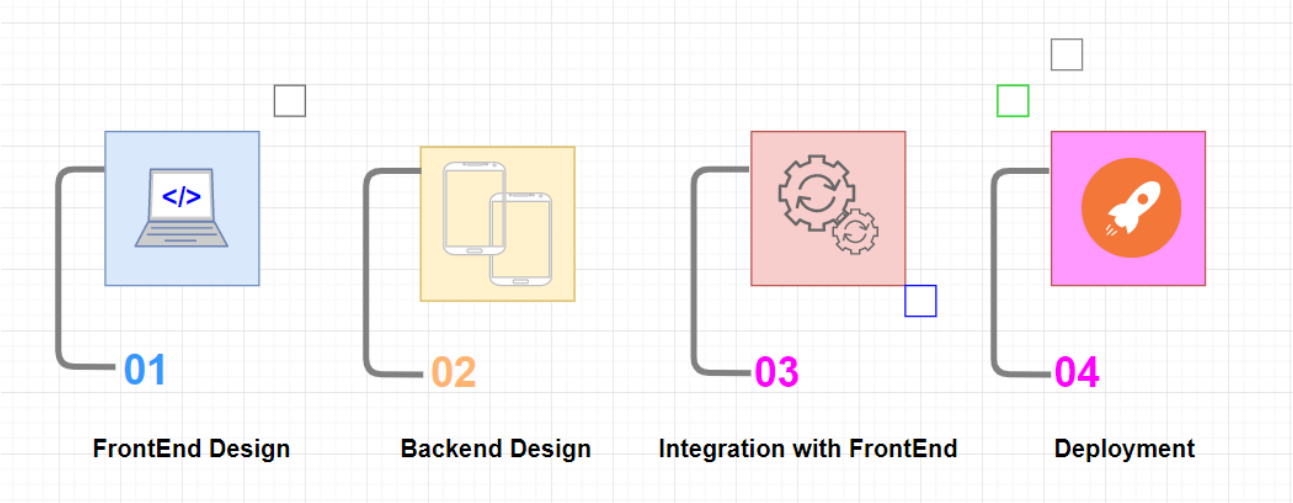


Figure 1: Project Flow

Chapter 2

Literature Reviews

2.1 Literature Reviews of some of the previous related studies are provided below:

Sl. No.	Paper Title	Authors	Publication Year	Objectives
01.	Automatic question generation and answer assessment for subjective examination[1].	Bidyut Das, Mukta Majumder, Arif Ahmed Sekh, and Santanu Phadikar.	2022	This project focuses on creating subjective questions, as well as a mechanism for evaluating the responses. Key terms from the course content are used to create the questionnaires (syllabus). Several forms of subjective questions are generated based on the keywords.
02.	Smart Paper Generator[2].	Anjali Sunil	2021	Teachers only need to upload the question bank for each subject to the smart paper generator. The question bank will be used to generate the question paper. This procedure is carried out automatically and with the assistance of a randomization algorithm.

03.	Designing an Adaptive Question Bank and Question Paper Generation Management System [20]	Pankaj Dwivedi, R. Tapan Shankar, B. Meghana, H. Sushaini, B. R. Sudeep & M. R. Pooja	2020	The design and the implementation of an automatic question paper creation and retrieval system for the engineering sector are discussed in this work. The administrator and login module, the question input module, the question retrieval module, and the evaluation module are the four modules that make up the system. A dynamic approach with low redundancy is used to construct the question paper. The question paper can be adjusted according to testing requirements, such as basic to advanced levels of difficulty because the entered question items are marked for their difficulty index. For objective questions, the evaluation module generates password-protected expert confirmed answer keys, while for subjective questions, it generates answer cues. The method may be able to meet the demand for confidential various sets of question papers with the same difficulty index for competitive engineering examinations or tests in a timely manner.

04.	Design and evaluation of an ontology-based tool for generating multiple-choice questions[3].	Cubric, M. and Tosic, M.	2020	The goal of this work is to explain and assess a tool built by the authors that produce test questions from any domain ontology using Bloom's taxonomy and strong pedagogical principles.
05.	Developing a Framework for Online Practice Examination and Automated Score Generation[4].	S. M. Saniul Islam Sani, Rezaul Karim, and Mohammad Shamsul Arefin	2019	Provide a framework that can take multiple-choice questions (MCQ) and written examinations. Create a database to record the questions and responses. The database's questions are shown on a web page, with MCQ questions having answer options and written questions having text boxes. Used different forms of analysis of the written questions' replies to generate the scores for the written questions. However, to calculate the MCQ question scores, it simply compared the database answers to the user's replies.

06.	Bloom’s taxonomy-based examination question paper generation system[5].	Yulia Timakova and Kinn Abass Bakon	2018	This work aims to develop an automated examination question paper generation system (AQPGS) to replace academics' manual methods. The system prototype was created in Visual Basic and connects to a Microsoft Access database. It has multiple-choice, True/False, and open-ended questions. Using a keyword query and a random selection of questions, a mapping algorithm is integrated for automated categorization of open-ended questions according to Bloom's Taxonomy hierarchy. The generated paper can be saved and changed as a text document.
07.	Design and realization of test question bank database System[6].	Xin Wanga, Zhong Wangb, Wei Huangc, Guanqi Wen, and Shaolei Zhangd	2017	Completes the test questions management, the examination paper management, and the student examination function which the examination question bank system request, but also has consummated and improved the function of the test question bank system.

08.	Proposed Bio-authentication System for Question Bank in Learning Management Systems [19].	Hussein Y. AbuMansour	2017	It provides a shared architecture for a Bio-authentication technique based on the authorized person's fingerprint as a nested internal security level for accessing the question bank. We feel it is an innovative approach to many common hacking scenarios, such as leaving authorized access with high privileges unattended for a variety of reasons, which makes it easy for unauthorized users to access the question bank. In the event of the first level's (Password/username) breaking, this unique approach is rapidly developing fingerprints as data in the internal information security progress of user authentication.
09.	Development and Validation of Web-Based Question Bank and Evaluation of Its Utility among Students and Teachers[7].	Parthasarathy, M. and Ananthasayanam, R.	2016	The purpose of this project was to create a Web-based Past Examination Question Bank (WPQB) for selected subjects with customized search capabilities, and the website was then reviewed to validate its usefulness among students and teachers.

10.	A taxonomy assessment and item analysis of a retailing management multiple-choice question bank[8].	John R. Dickinson	2015	Banks of multiple-choice questions, and the taxonomies into which the questions are classified, e.g., by difficulty and question type.
11.	Medical school 2.0: How we developed a student-generated question bank using small group learning[9].	Adrian C.C. Gooi; and Connor S. Sommerfeld	2014	Students produced their own multiple-choice questions (MCQs) using self-study tools, and then reviewed each other's questions in small groups. Selected questions were discussed with the entire class. The instructor then reviewed all of the questions and added them to a question bank that students may use for formative learning.
12.	An algorithm for question paper template generation in question paper generation system[10].	Vaibhav M. Kale, and Arvind W. Kiwelekar.	2013	We propose the creation of an algorithm to generate a question paper template that meets the aforementioned conditions in this paper. The algorithm is demonstrated in the paper utilizing four restrictions based on Bloom's taxonomy: question paper style, syllabus coverage, difficulty level coverage, and cognitive level coverage.

13.	Design of adaptive question bank development and management system[11].	Vijay Krishan Purohit; Abhijeet Kumar; Asma Jabeen; Saurabh Srivastava; R H Goudar; Shivanagowda; and Sreenivasa Rao	2012	Developing an adaptive question bank management system that automatically selects questions from a large database (question bank) and represents the question model based on the question paper designer's inputs or criteria (QPD). The question modeling process will be ensured by the idea map combined with the question bank (question database) based on the degree of specific criteria such as Bloom's Taxonomy, difficulty level, and so on.
14.	A framework for automated generation of examination questions from web based semantically treated search results	Horst Liske	2011	The paper discusses a programming framework for creating queries from the internet help resources automatically. It also provides tips and recommendations to help you locate the proper solution.

Chapter 3

Technology

3.1 Introduction

The technologies that have been used in this project include ReactJS for Front-End design. It is very flexible and easy to make interactive web pages with React. We also use Redux and Node.js Express. Material UI has been used for page design. MVC Framework has also been used. And MariaDB for the backend database.

3.2 Description

3.2.1 ReactJS

React is an open-source as well as flexible, and declarative JavaScript library that is easy to use for developers to build interactive, scalable, simple, and fast frontend interfaces for single-page or multi-page web apps. React makes it very easy for developers to create attractive UIs. It creates basic views for every state of our project, and React will render and update the appropriate components as our data changes. Declarative views of it improve the predictability and the debuggability of the code. React composes encapsulated components that handle their own state to create complex and complicated user interfaces. We can simply transmit rich data through your app and keep the state out of the DOM since component logic is written in JavaScript rather than templates. Because we don't make assumptions about the rest of your technology stack, you can use React to build new features without having to rewrite old code. Node.js can also be used in React to render on the server and React Native to power the mobile applications.

We use React instead of HTML and CSS because –

- React has been seen as being faster than HTML. We can use declarative HTML syntax directly in the JavaScript code with ReactJS.
- It allows us to create separate and smaller code components and files that are easy to execute.
- React creates a more responsive UI (user interface) that is very easy for developers.
- It adds dynamic features to the project using JavaScript, which is very helpful.
- React is not only good for its performance, but also its clear design is highly test-friendly, which means the applications are highly testable. It can be easily supervised from the functions, triggered outputs, and events etc.

- We just have to import React libraries. JSX is a special syntax that looks like HTML and converts the API call of React, and it renders the HTML.

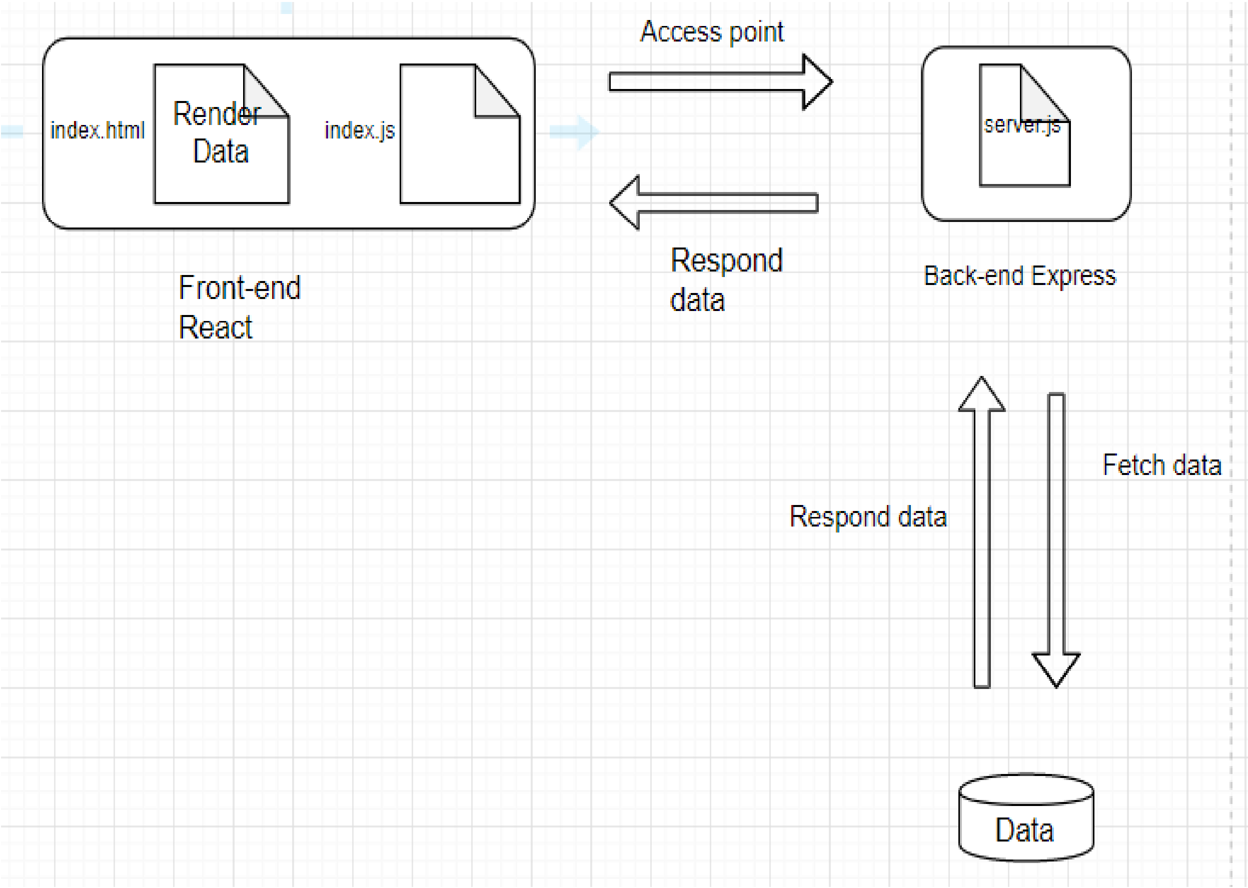


Figure 2: React Architecture

3.2.1.1 Virtual Document Object Model (DOM) :

React's lightweight counterpart of the Real DOM is the Virtual DOM. Real-world DOM manipulation takes much longer than virtual DOM manipulation. Virtual DOM just updates that object in the real DOM when its state changes, not all of them. VDOM is updated when the state of an object in a React application changes. Then, rather than updating all of the items in the actual DOM, it compares its prior state and changes only those in the real DOM. Things can move more swiftly as a result of this, especially when compared to other front-end technologies, which must update each item even if just one object in the web application changes.

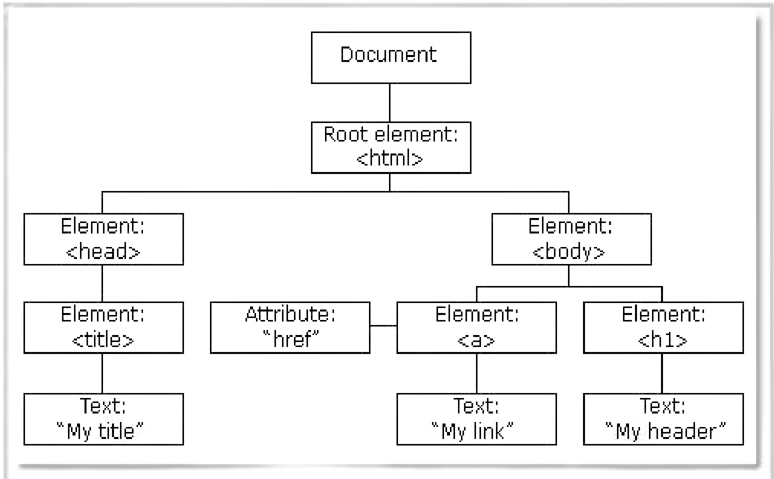


Figure 3: DOM of a Webpage

3.2.1.2 React-router

React Router is a navigation library for React applications. React-router-dom and React-router-native are both included in this collection. The first is for web application navigation, and the second is for mobile application navigation (React Native). It will be possible to traverse the web application from one page to another using this library, regardless of where we are in the application. It will be possible to construct routes and then follow one of them using the link.

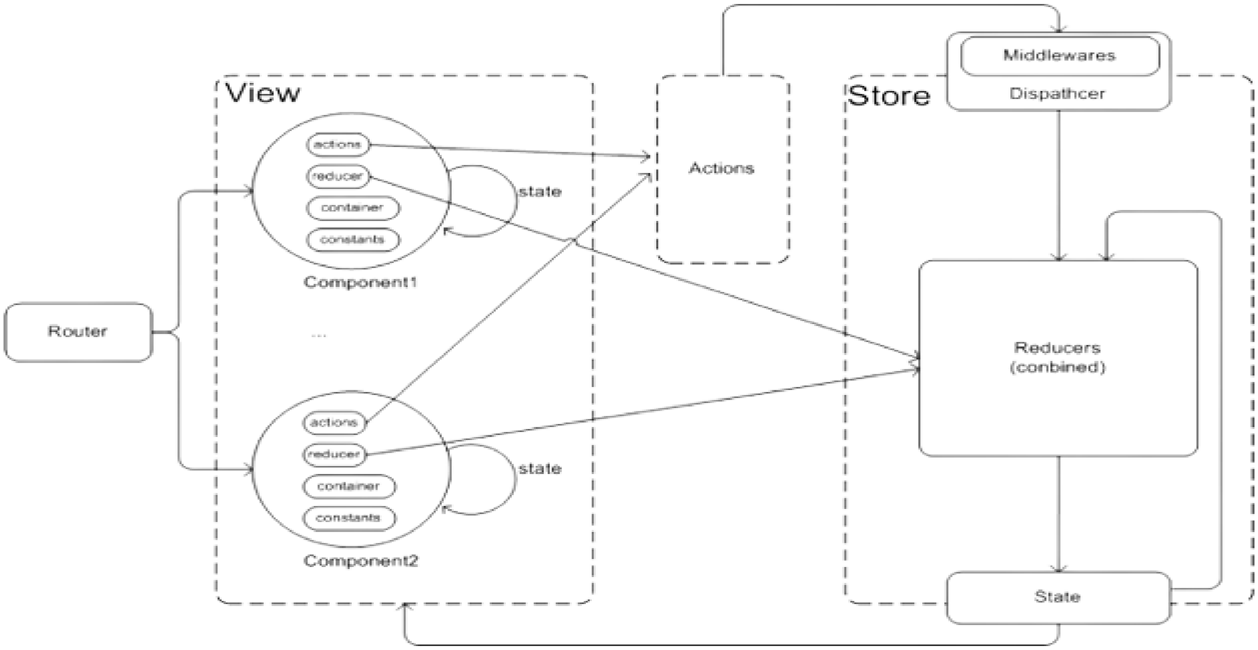


Figure 4: React Router

This is a UI MVC architecture chart. The view is a React class component with its own states, constants, actions(events), reducers (event handlers), and containers (connect to the Redux global store). The model and controller are the Redux store, which acts as a global centralized manager, dispatching actions and executing reducers. The state change will, in turn, result in the React component being updated.

3.2.2 Redux

Redux enables you to create apps that act consistently across environments (client, server, and native) and are simple to test. The ability to centralize your application's information and logic allows for powerful features like undo/redo, state persistence, and more. The Redux tool makes it very simple to see where, when, why, and how the state of our application has been changed. The design of Redux allows us to notice the changes, "time-travel debugging," and it even sends total fault reports to the server. It can be used with any UI layer, and Redux has a wide ecosystem of addons to customize it as per our requirements. Redux aids app scaling by offering a logical mechanism to handle the state via a one-way data flow architecture. The React Redux concept is very clean and straightforward. It joins the Redux store, checks if the data our component requires has changed, and then re-renders our component.

We use React-Redux because -

- The official UI bindings for the React applications are known as React-Redux. It's kept up to date with any API updates to guarantee that your React components work as they should.
- It promotes the use of React architecture.
- Many speed improvements are implemented internally, allowing components to re-render only when necessary.

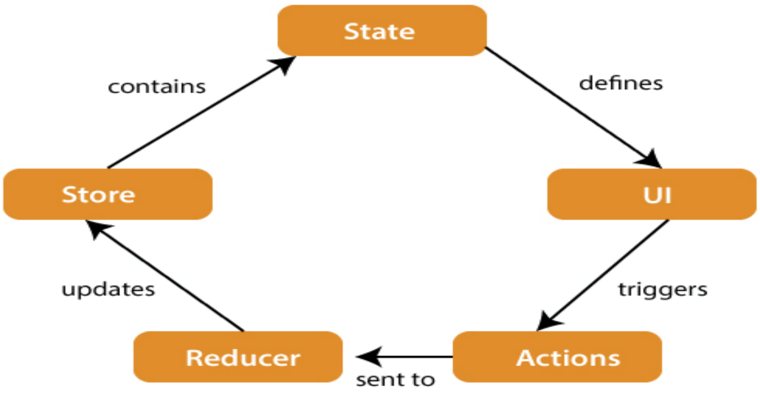


Figure 5: React Redux Architecture

3.2.3 Material UI

Material is a flexible set of components, guidelines, and tools that enable user interface design very easily, and best practices. Material Design is a design language developed by Google for Android that supports touchscreen interactions with cue-rich features and natural movements that mirror real-world items. Material Design has been widely accepted by the design community, and it can now be seen widely on websites and applications that aren't built by Google. In simple words, the Material Design is very efficient now for both desktop and mobile applications. Material-UI is a modern package that allows developers to use and import multiple components efficiently in their React apps to construct a user interface. As the developers do not have to rewrite everything from the beginning, this saves a lot of time for the project. Material-UI widgets are highly influenced by Google's user interface design concepts. As a result, it is simple for developers to create aesthetically appealing apps. Currently, the integration of the Material UI library and React.js projects has whole thing to a very new level. Some frontend frameworks are poorly documented, making it difficult to work with them. Material UI, on the other hand, provides extensive documentation that makes it simple to traverse through the framework. Material UI is updated on a regular basis to keep it current. Its components are similar in design and colour tones, which result from an aesthetically pleasing web application or any webpage.

3.2.4 MVC Framework

The Model-View-Controller (MVC) architecture pattern has three logical components: model, view, and controller, and it divides any application development into these parts. Each of these three components is designed to handle the specific parts of application development. MVC is an industry-standard, very popular web development framework in modern technologies for developing scalable and flexible projects. All the data-related logic that the user engages with is represented by the Model component. This might be the data that is being transmitted between the View and Controller components or any other data related to the business logic. A Customer object, for example, will get the customer information from the database, change it, and either update or output the data back to the database. All of the UI logic of the applications is handled by the View component. The Customer view, for example, will construct all of the UI components that the final user interacts with, such as text fields, dropdowns, and so on. The controllers serve as the link between Model and View components, processing all business logic and incoming requests, manipulating data using the Model, and interacting with Views to produce the final output. The customer controller, for example, will handle all the inputs and interactions from Customer View and will use the Customer Model to update the database. And Customer data can be seen using this same controller.

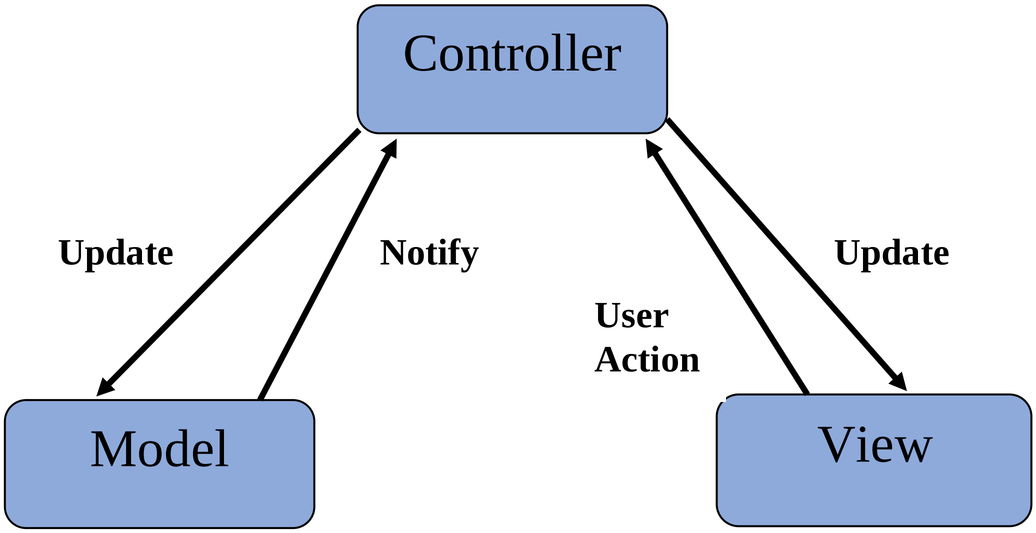


Figure 6: MVC Framework

3.2.5 MariaDB

MariaDB is an open-source relational database management system (DBMS) that may be used as a drop-in replacement for the popular MySQL database. It was built as a MySQL software fork by the people who were engaged in the development of the original database. It is a SQL-based database that allows ACID-style data processing with assured atomicity, consistency, isolation, and durability. The database also supports JSON APIs, concurrent data replication, and various storage engines, among other things. It is the most powerful open-source relational database, with support for current SQL and JSON, Oracle Database compatibility, high availability, and robust security. It is a database that may be used as a regular database, a distributed SQL database, or a data warehouse. In a wide range of applications, from banking to the internet, it converts the data into structured information. MariaDB is utilized because it is fast, scalable, and resilient, with a rich ecosystem of storage engines, plugins, and other tools that make it highly adaptable for a wide range of use cases. It was originally created as the upgraded, drop-in replacement for the MySQL database. It is the relational database that uses SQL interfaces to retrieve the data. It has been created as open-source software. GIS and JSON capabilities have been added to MariaDB in recent editions.

3.2.6 Node.js Express

Express is a Node.js web application framework that offers a comprehensive range of functionality for both online and mobile apps. Using a variety of HTTP utility methods and middleware, you can quickly and easily build a powerful API. Express adds a thin layer of basic web application functionality without obscuring the Node.js capabilities you already know and love. Express is the foundation for several prominent frameworks. Express is a Node.js web application framework that includes a wide range of features for developing web and mobile apps. It makes it easier to create Node-based Web apps quickly.

Some of the popular key features of the Express framework are listed below:

- It allows middleware to reply to HTTP requests to be built up.
- It defines a routing table for doing various actions based on HTTP Method and URL.
- It allows us to render the HTML pages dynamically by supplying the variables to the templates.

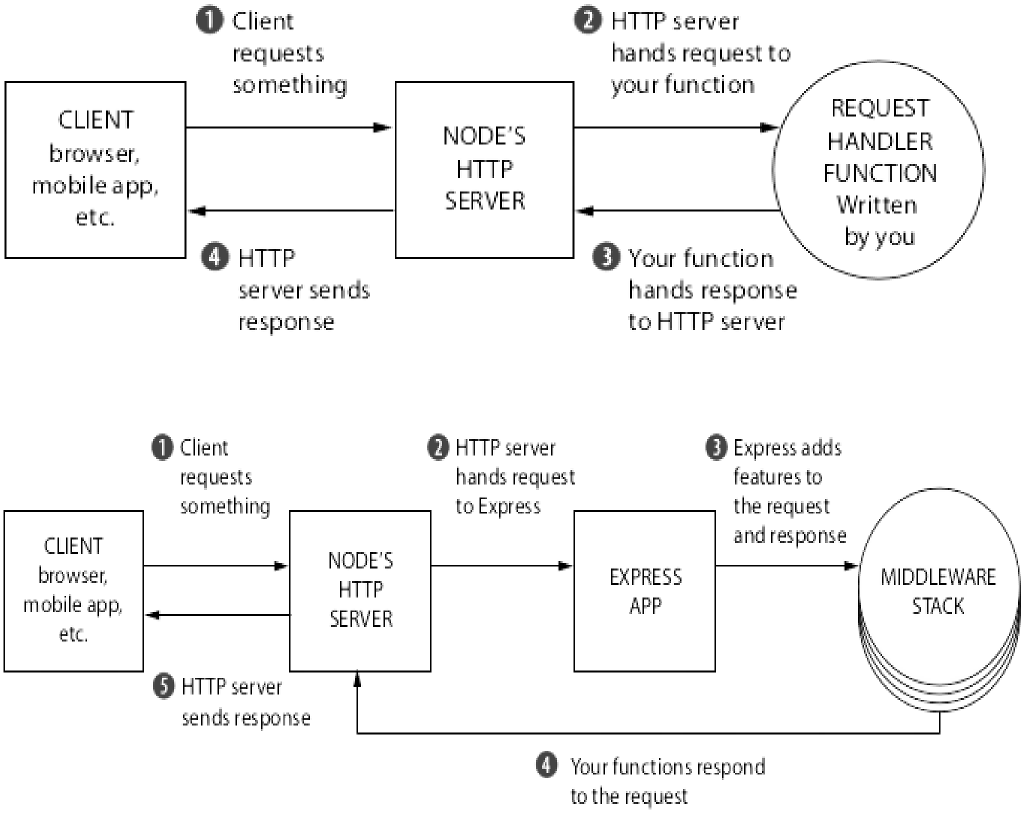
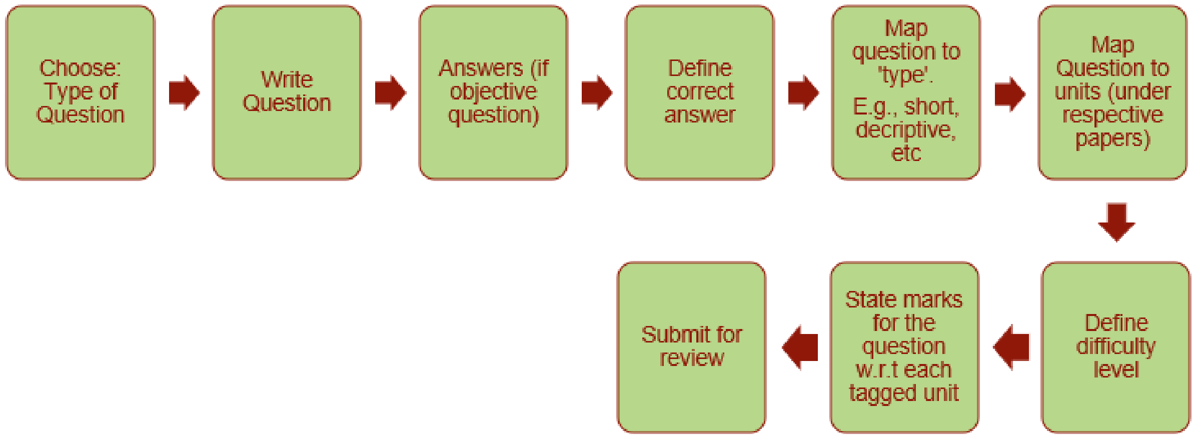

❶ Client requests something
❷ HTTP server hands request to your function
CLIENT browser, mobile app, etc.
NODE'S HTTP SERVER
REQUEST HANDLER FUNCTION Written by you
❹ HTTP server sends response
❸ Your function hands response to HTTP server
❶ Client requests something
❷ HTTP server hands request to Express
❸ Express adds features to the request and response
CLIENT browser, mobile app, etc.
NODE'S HTTP SERVER
EXPRESS APP
MIDDLEWARE STACK
❺ HTTP server sends response
❹ Your functions respond to the request

Figure 7: Node.js Express

Chapter 4

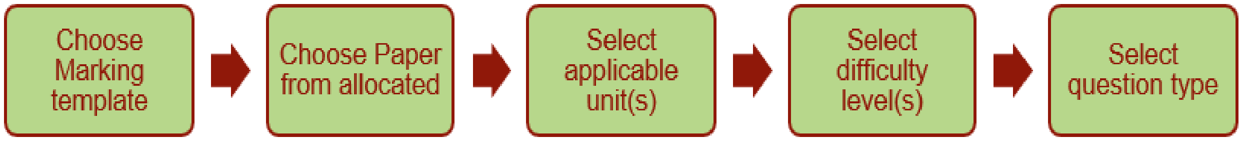
Methodology

4.1 Questions Creation :

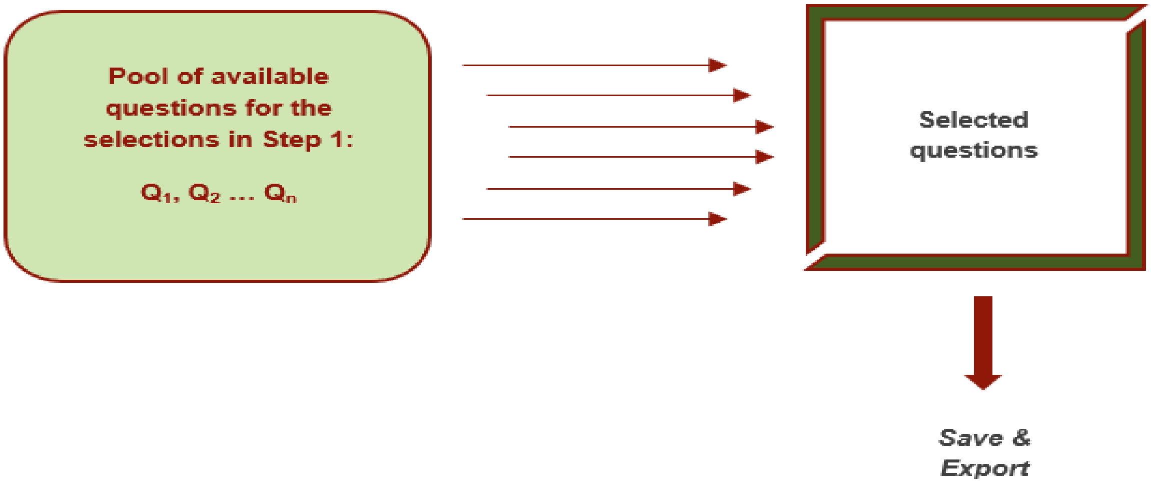

Choose: Type of Question
Write Question
Answers (if objective question)
Define correct answer
Map question to 'type'. E.g., short, decriptive, etc
Map Question to units (under respective papers)
Define difficulty level
State marks for the question w.r.t each tagged unit
Submit for review

4.2 Creating Question Paper

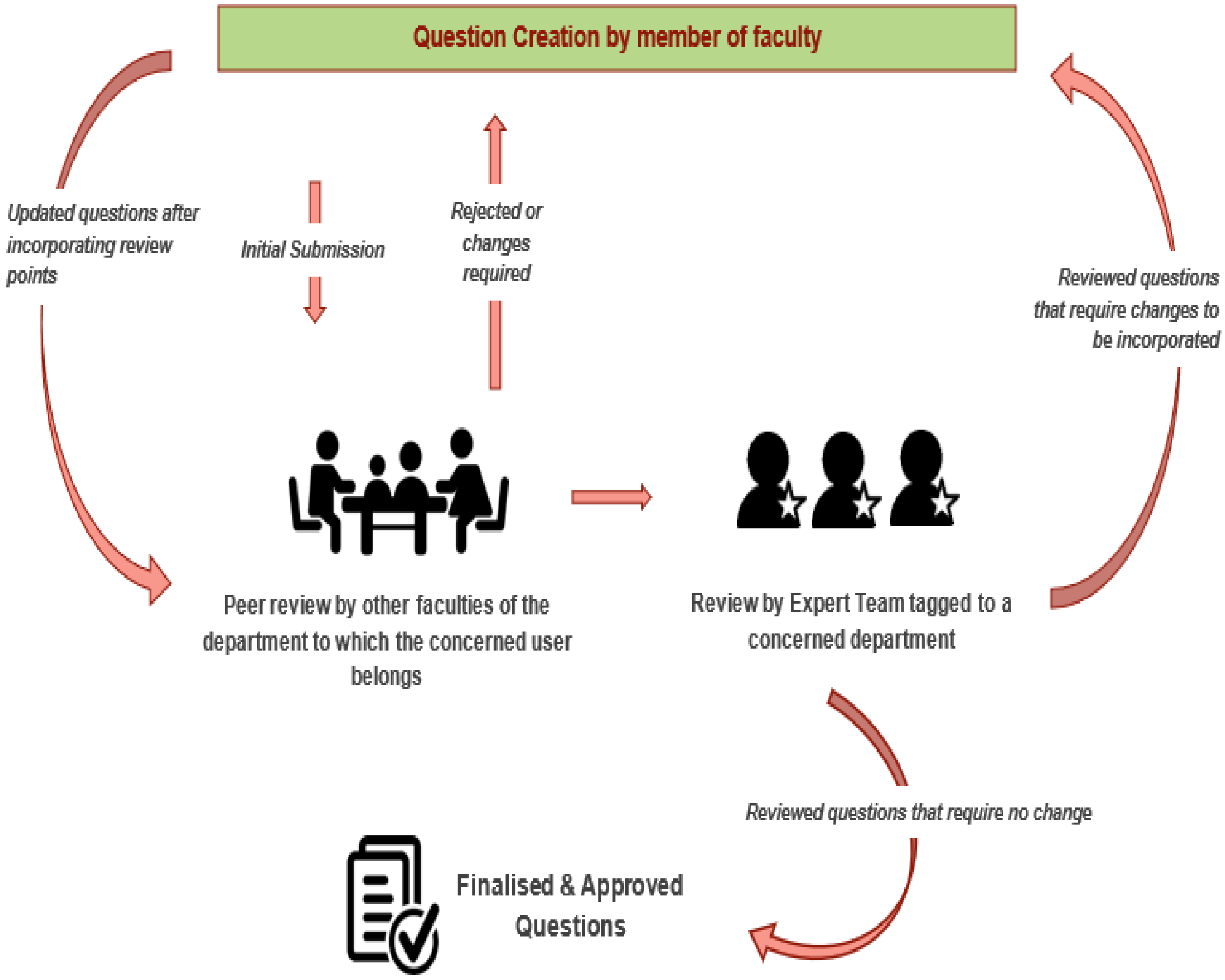
Step – 1


Choose Marking template
Choose Paper from allocated
Select applicable unit(s)
Select difficulty level(s)
Select question type

Step - 2



4.3 Review Process :


Question Creation by member of faculty
Updated questions after incorporating review points
Initial Submission
Rejected or changes required
Reviewed questions that require changes to be incorporated
Peer review by other faculties of the department to which the concerned user belongs
Review by Expert Team tagged to a concerned department
Reviewed questions that require no change
Finalised & Approved Questions

It supports three types of users - teachers, reviewers, and admins. Two different types of Dashboards will open for teacher’s login and reviewer login on the other side. After successful log-in or sign-up teachers can create questions by selecting the type of the questions and the difficulty level of the questions. Then they can create the question paper subject-wise. Then the question papers should be reviewed and approved by the other faculty members of the respective department of the institute. Once they will approve the questions set, the questions will be published.

Chapter 5

Software and Hardware requirements

5.1 Software and Hardware:

The software and hardware requirements of this project are –

- Operating System: Windows 7 and above
- Language: Html, CSS, JavaScript
- Technology Stack:
    1. MariaDB,
    2. ReactJS,
    3. Node.js Express,
    4. MVC Framework,
    5. Material UI
- Browser: Any browser and IE 8 and above.
- Database Language: MariaDB
- Processor: A single-core 2GHz processor
- RAM: 512 Mb and above

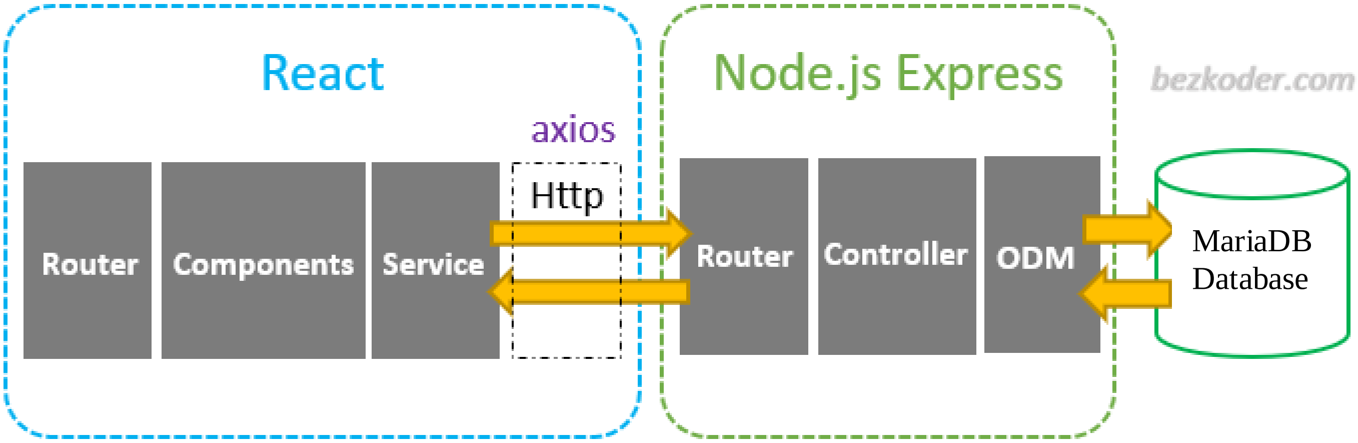


Figure 8: Software requirements

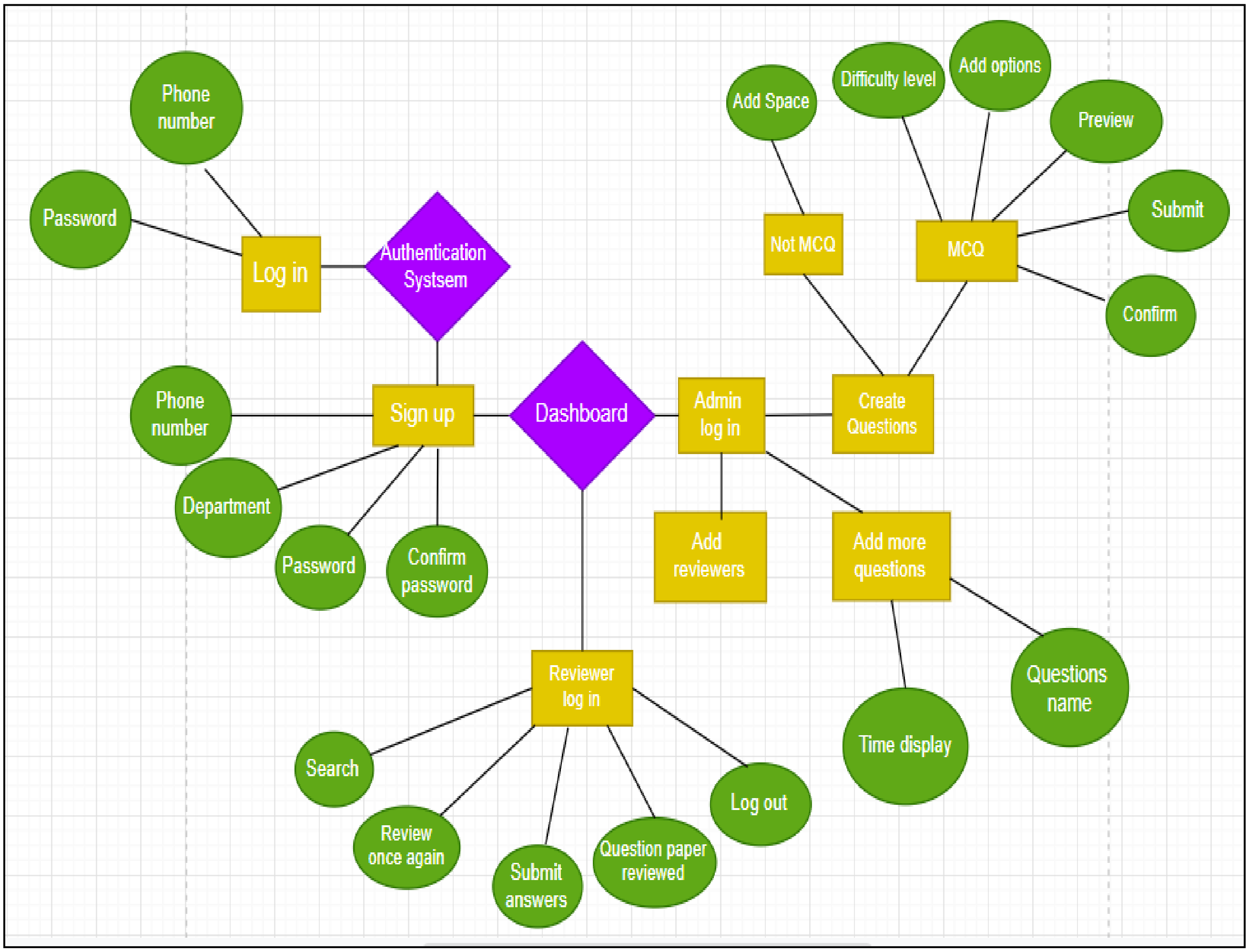
Chapter 6

Implementation and results

6.1 ER Diagram:

An entity-relationship diagram (ERD), often called an entity-relationship model, is a graphical depiction of relationships between things, people, locations, concepts, events, etc, in an information technology (IT) system. ER diagrams are widely used to represent and make relational databases, both in terms of logic and business rules (in the logical data model) and the specific technology to be employed (in the physical data model.) Real-world things are represented using an ER model.

Figure 9: ER Diagram for Smart Question Bank



6.2 Description:

Step 1:

First, there is an authentication system for any person. It will show two options –

- Log in
- Sign up

Step 1.2:

If sign up has been chosen, it will ask for –

- Phone number
- Department
- Password
- and finally confirm the password

And if login is chosen, it will ask for only

- Phone number
- Password

Step 2:

Then the Dashboard page will open.

There may be three types of people who log in

- Admin Log in
- Reviewer Log in
- Teacher login

Step 2.1:

For teachers, they can see the following options -

- Add reviewers
- Create Questions
- Add more questions
- Level of the difficulty

Step 3:

In the ‘Create Question’ section, they can have the following parts -

- Create questions

- Then Submit
- Preview
- Confirmation button

Then they will be back to the Dashboard page.

Step 4:

In the ‘Add more Questions’ section, there will be the following options -

- Question name

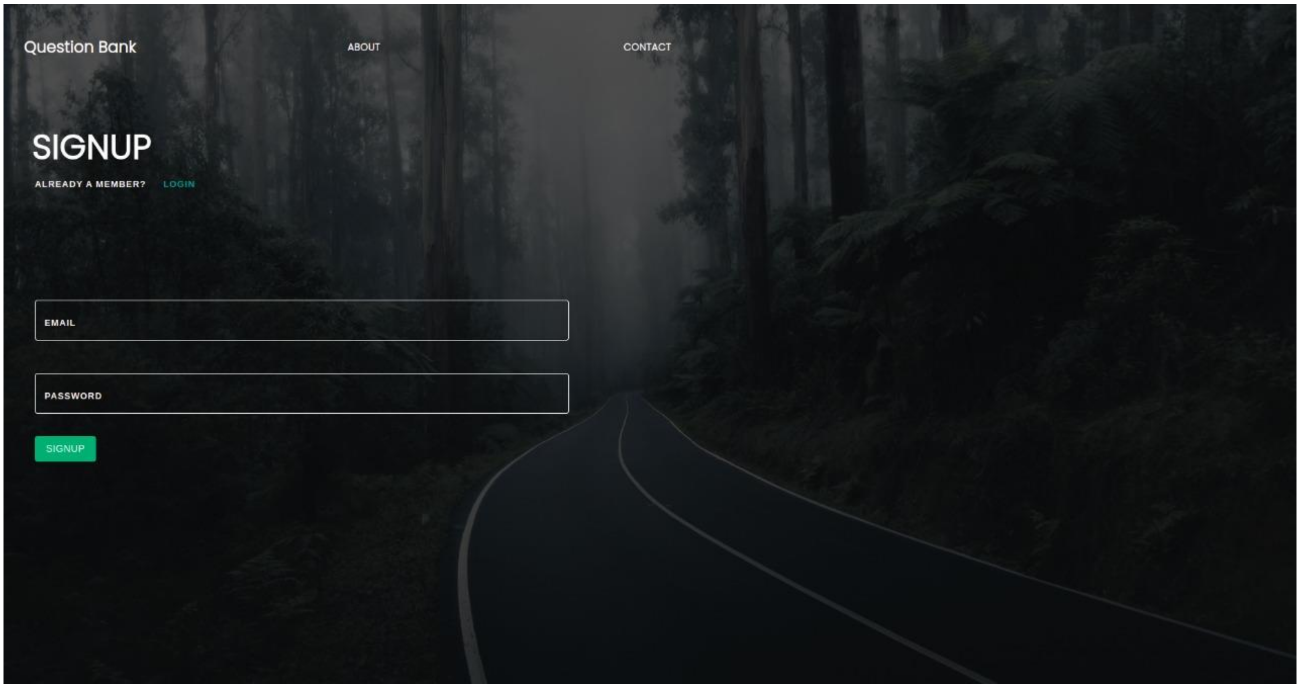
Step 5:

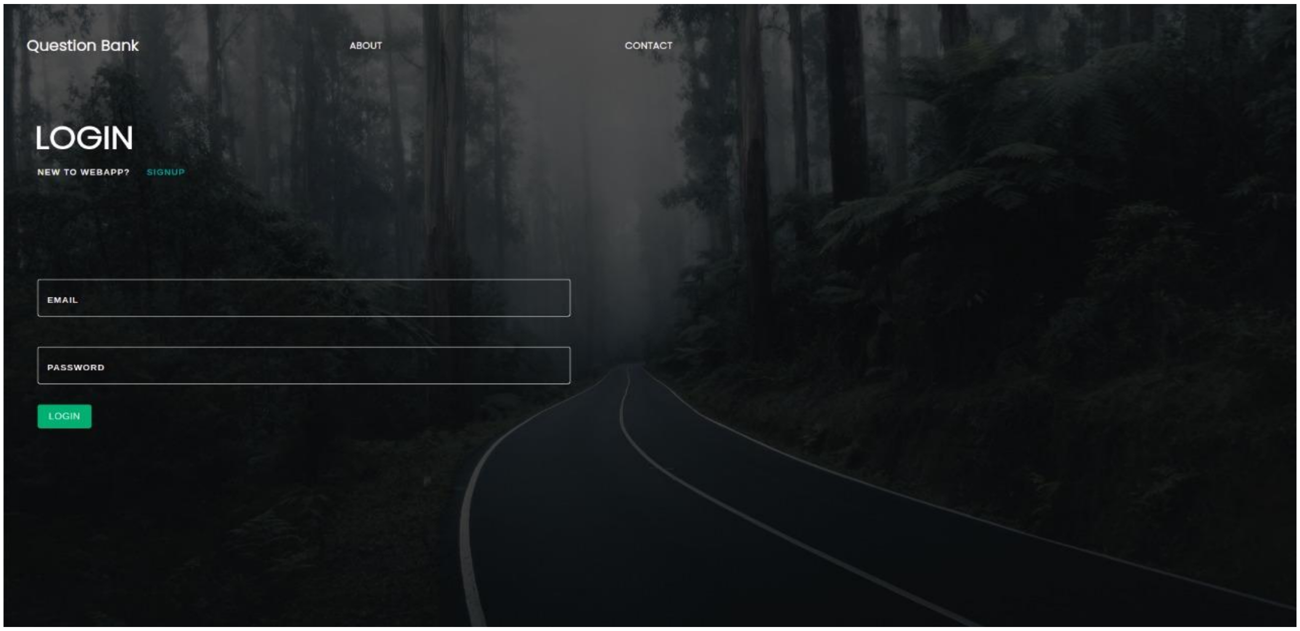
And for the reviewer to log in, they can view the different dashboards. They can have the following options –

- Search button
- Review once again
- Log out

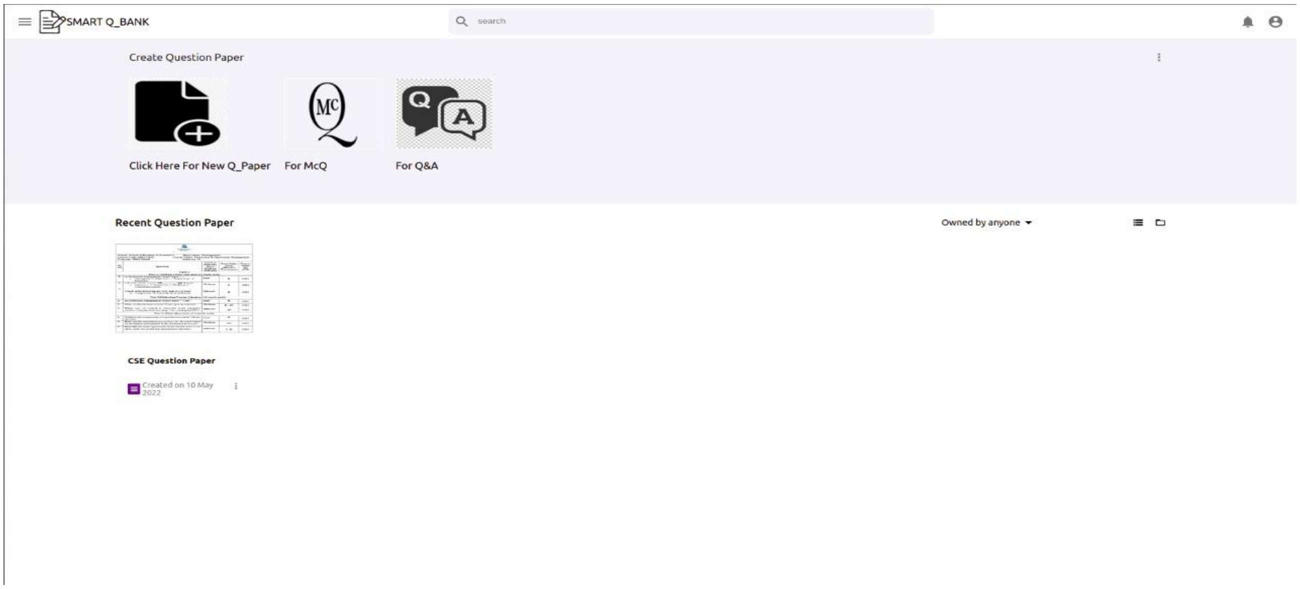
6.3 Results

Sign-On feature with the help of login (authentication)





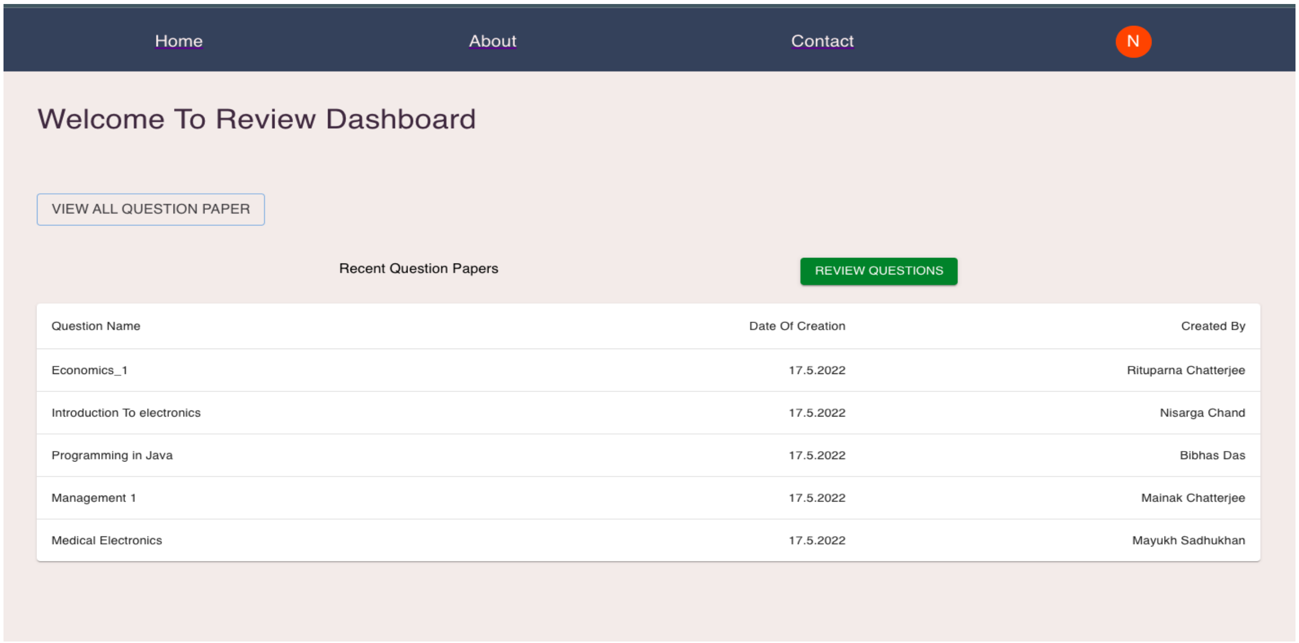
Support the following user roles:



Contributor

Support the creation of the following question pattern:

- MCQ
- General question



Welcome To Review Dashboard

Recent Question Papers

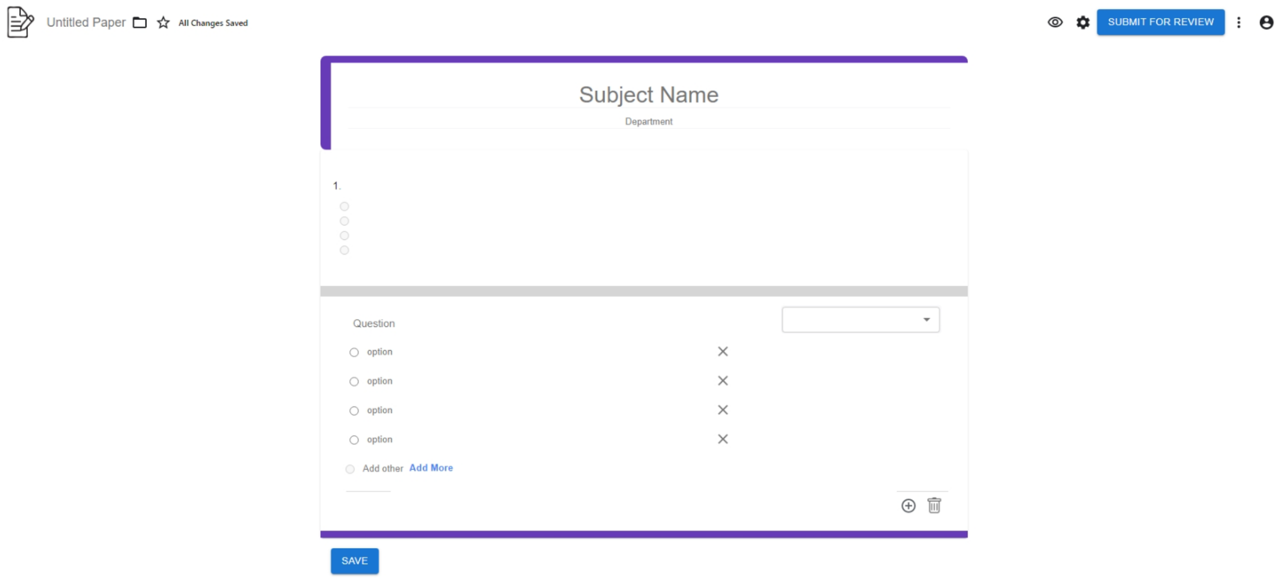
Question Name	Date Of Creation	Created By
Economics_1	17.5.2022	Rituparna Chatterjee
Introduction To electronics	17.5.2022	Nisarga Chand
Programming in Java	17.5.2022	Bibhas Das
Management 1	17.5.2022	Mainak Chatterjee
Medical Electronics	17.5.2022	Mayukh Sadhukhan

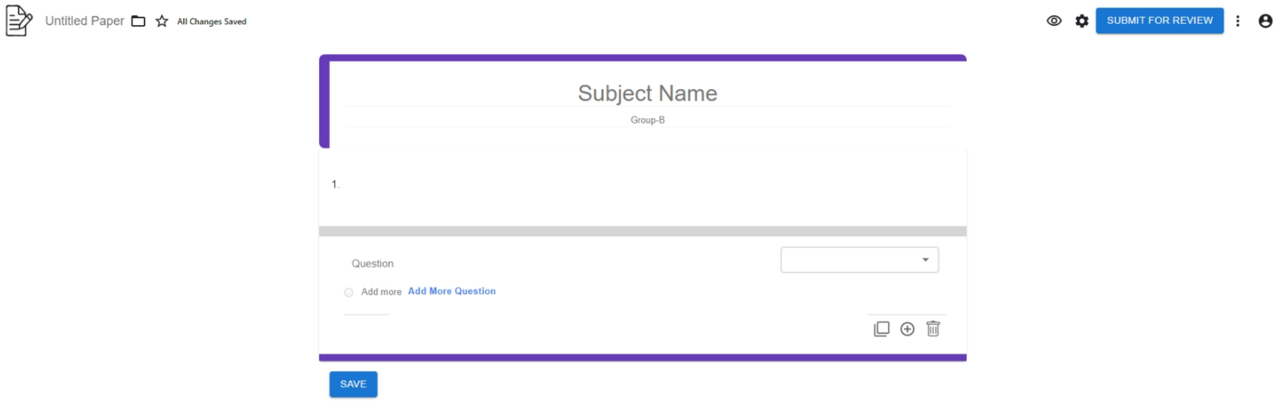
Reviewer

Peer & expert review for questions submitted by other members of faculty belonging to the concerned academic department.

For teachers –

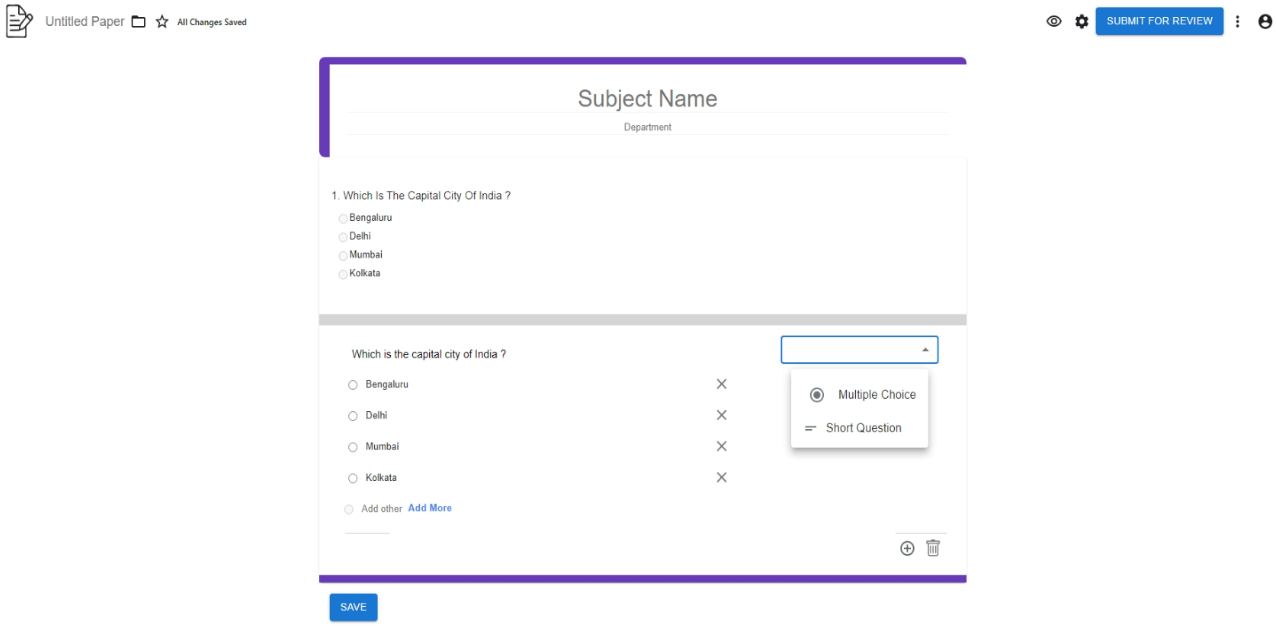
- Create Questions
- Add more questions





Map questions to the following type:

- Objective
- Short Questions
- Explanation-Based Questions



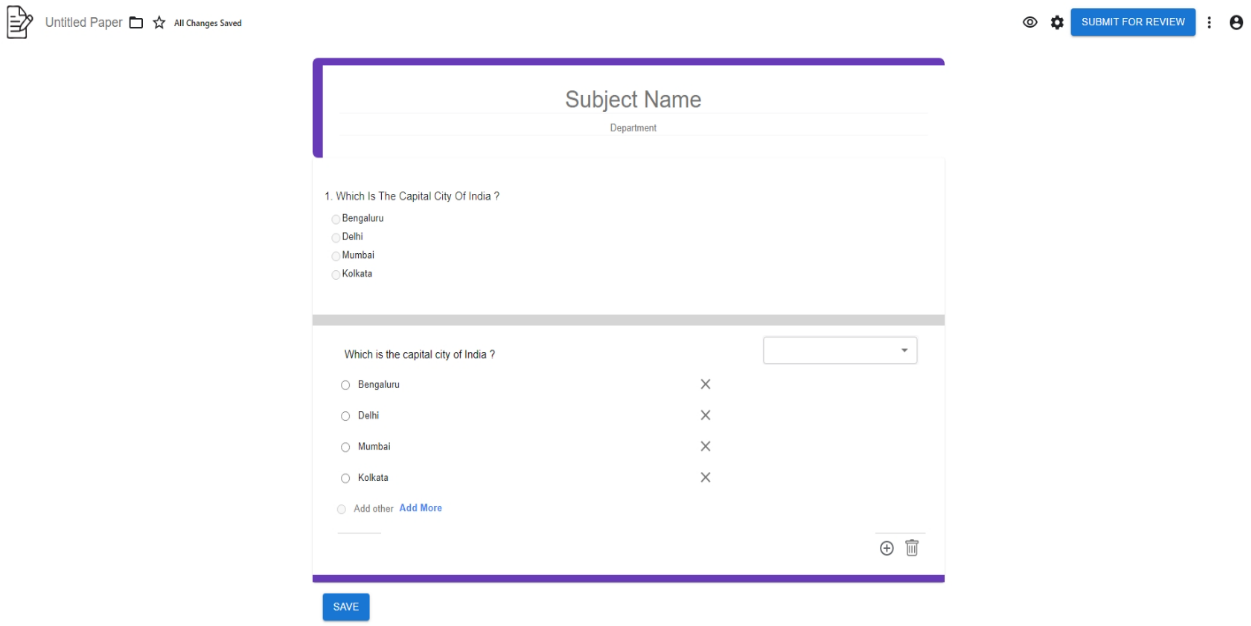
1. Which Is The Capital City Of India ?

○ Bengaluru
○ Delhi
○ Mumbai
○ Kolkata

Which is the capital city of India ?

○ Bengaluru
○ Delhi
○ Mumbai
○ Kolkata

- Support questions to be tagged to multiple papers/subjects belonging to different academic programs.



Subject Name

Department

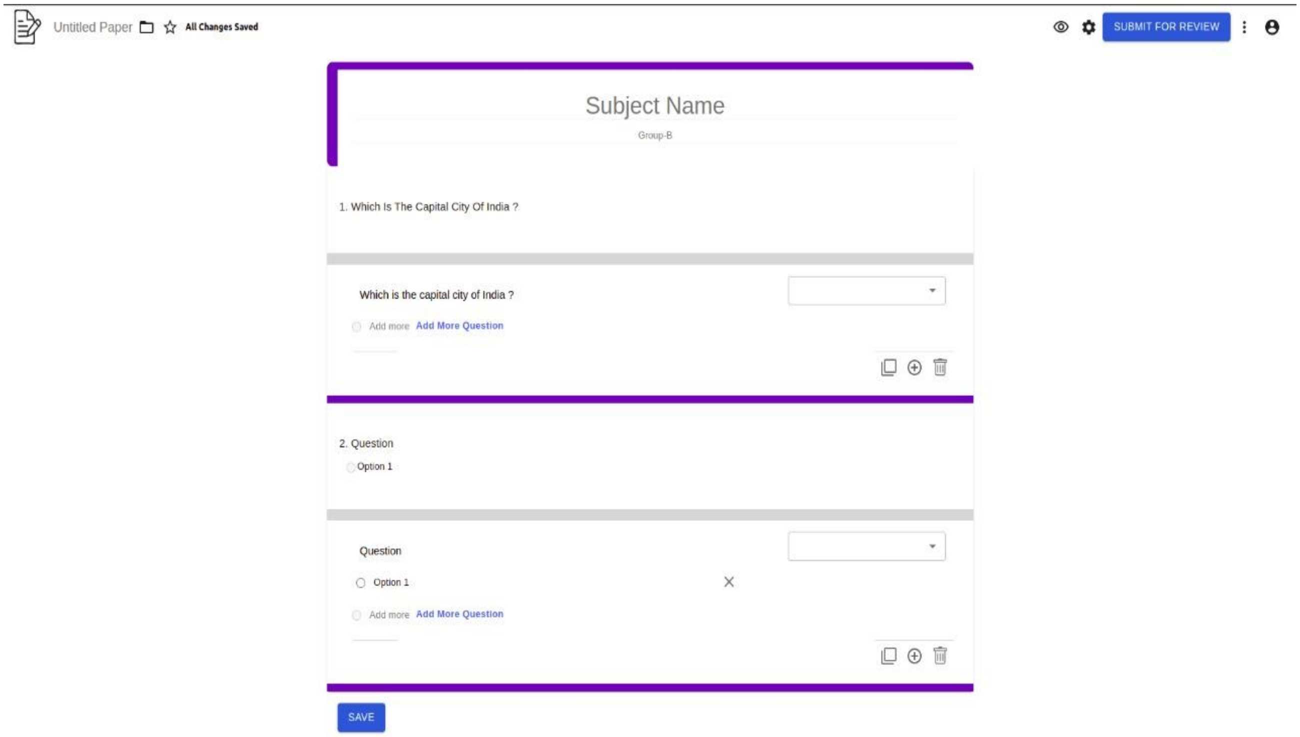
1. Which Is The Capital City Of India ?

○ Bengaluru
○ Delhi
○ Mumbai
○ Kolkata

Which is the capital city of India ?

○ Bengaluru ✕
○ Delhi ✕
○ Mumbai ✕
○ Kolkata ✕
○ Add other

For MCQ Types



Subject Name

Group-B

1. Which Is The Capital City Of India ?

Which is the capital city of India ?

○ Add more

2. Question

○ Option 1

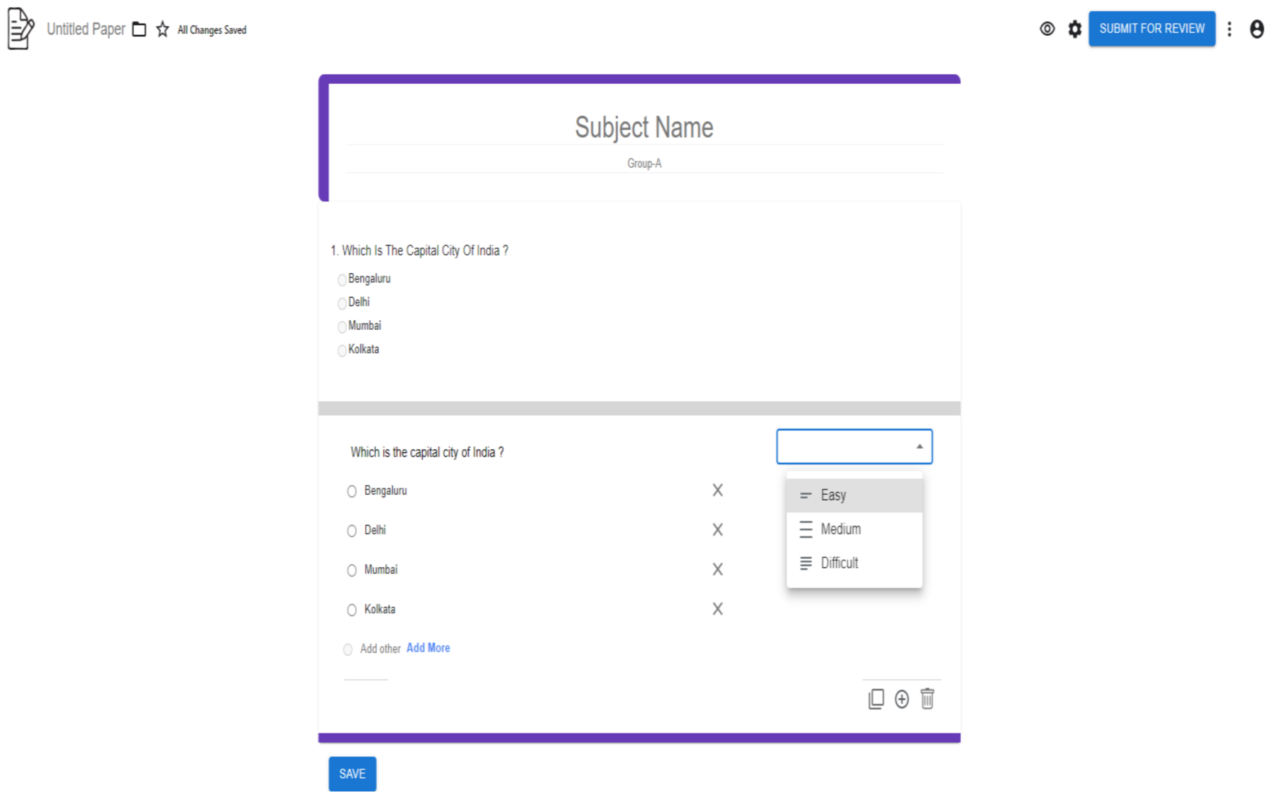
Question

○ Option 1 ✕
○ Add more

For Subjective Types

Support questions for the following difficulty level:

- Easy
- Medium
- Hard



Subject Name

Group-A

1. Which Is The Capital City Of India ?

○ Bengaluru

○ Delhi

○ Mumbai

○ Kolkata

Which is the capital city of India ?

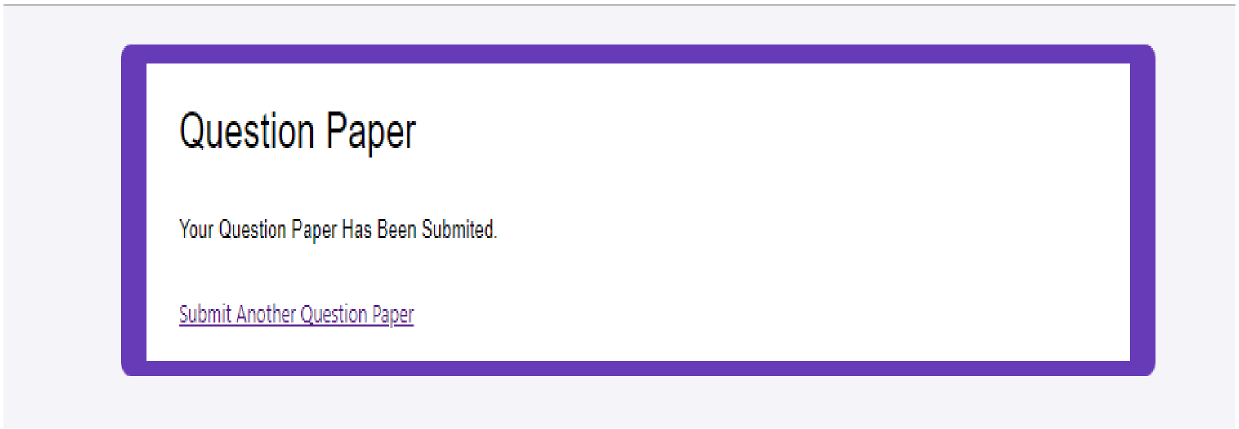
○ Bengaluru

○ Delhi

○ Mumbai

○ Kolkata

- After successfully submitting a question, this notification will pop up

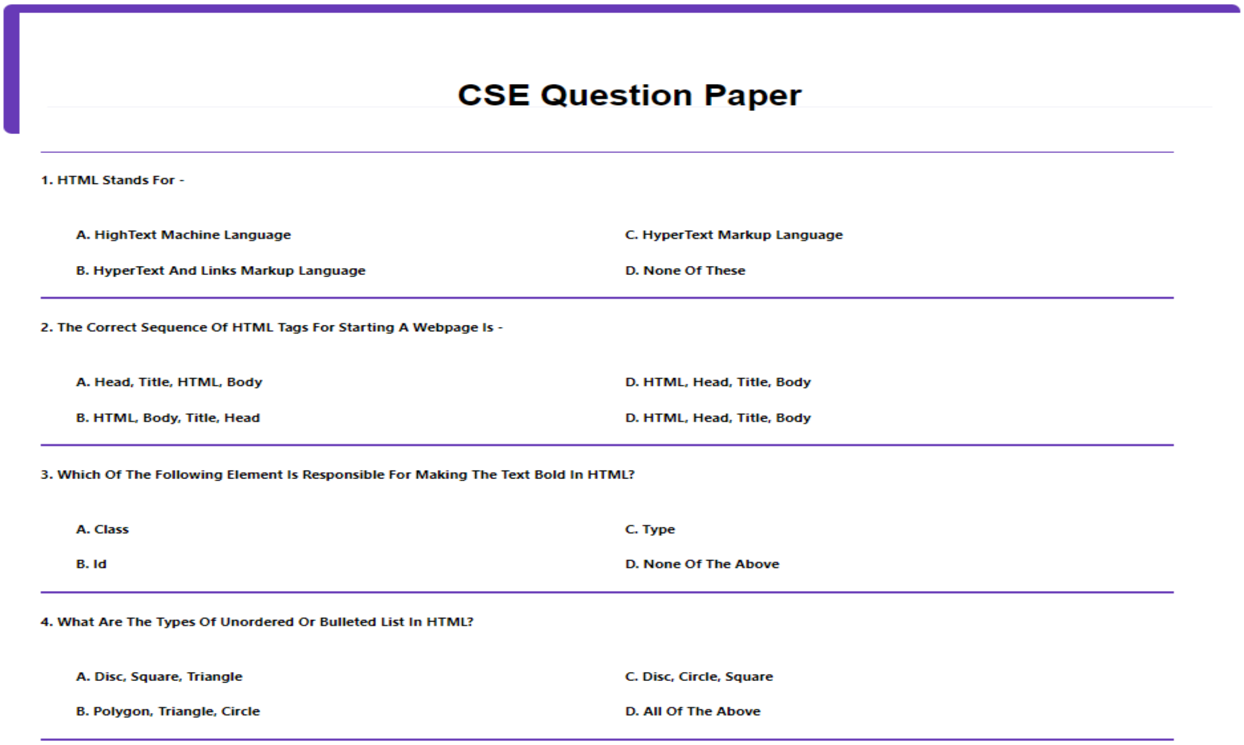


Question Paper

Your Question Paper Has Been Submited.

Submit Another Question Paper

- The preview page after submitting the question for review (MCQ), and the teacher can view it.



CSE Question Paper

1. HTML Stands For -

A. HighText Machine Language

B. HyperText And Links Markup Language

C. HyperText Markup Language

D. None Of These

2. The Correct Sequence Of HTML Tags For Starting A Webpage Is -

A. Head, Title, HTML, Body

B. HTML, Body, Title, Head

D. HTML, Head, Title, Body

D. HTML, Head, Title, Body

3. Which Of The Following Element Is Responsible For Making The Text Bold In HTML?

A. Class

B. Id

C. Type

D. None Of The Above

4. What Are The Types Of Unordered Or Bulleted List In HTML?

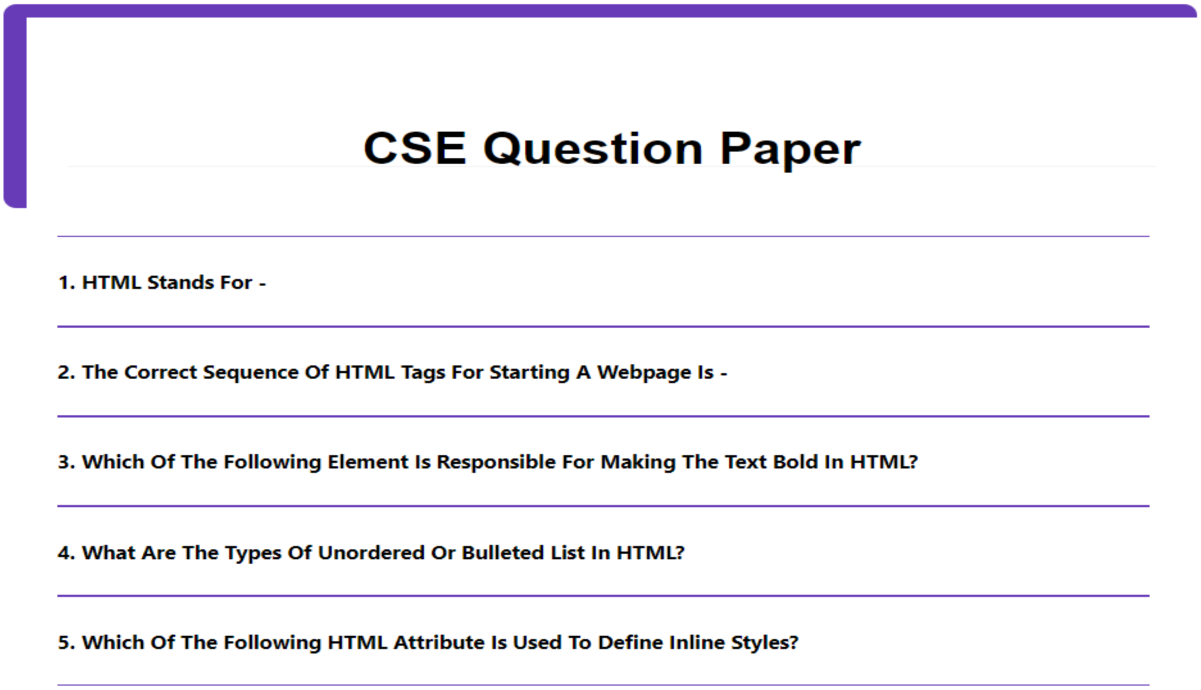
A. Disc, Square, Triangle

B. Polygon, Triangle, Circle

C. Disc, Circle, Square

D. All Of The Above

- The preview page (Description types) after submitting the question for review, and the teacher can view it.



CSE Question Paper

1. HTML Stands For -

2. The Correct Sequence Of HTML Tags For Starting A Webpage Is -

3. Which Of The Following Element Is Responsible For Making The Text Bold In HTML?

4. What Are The Types Of Unordered Or Bulleted List In HTML?

5. Which Of The Following HTML Attribute Is Used To Define Inline Styles?

Chapter 7

Conclusion

In this project, we have created a web platform where teachers can create several questions, like multiple-choice questions, long-answer type questions, etc., for checking the subject-wise preparation of students. From a proper analysis of the positive points and constraints on the component and research, it can be safely concluded that this product is highly efficient for creating questions. This application will work properly and meet all user requirements. This component can be easily accessible to the faculty, reviewers, and admins. Most of the existing solutions do not estimate cost. So, in all ways, we can conclude that this project is viable both technically and economically, and it will have high IT demand.

Chapter 8
Future Work

We will do the basic Artificial Intelligence (AI) implementation in the near future, which will make the whole process easier to use. In the future, we will create question papers according to the matrix. And we will include Bloom’s taxonomy in this project.

We will research the users of this web application along with their demands and any problems they are facing, and we will improve it and make it more efficient.

Chapter 9

Reference

1. Das, B., Majumder, M., Sekh, A. A., &amp; Phadikar, S. (2022). Automatic question generation and answer assessment for subjective examination. Cognitive Systems Research, 72, 14–22. https://doi.org/10.1016/j.cogsys.2021.11.002
2. Anjali Sunil. 2021. “Smart Paper Generator”. International Journal of Progressive Research in Science and Engineering 1 (9):17-19. https://www.journals.grdpublications.com/index.php/ijprse/article/view/222.
3. Cubric, M. and Tosic, M. (2020), "Design and evaluation of an ontology-based tool for generating multiple-choice questions", Interactive Technology and Smart Education, Vol. 17 No. 2, pp. 109-131. https://doi.org/10.1108/ITSE-05-2019-0023
4. Sani, S. M. Saniul Islam and Karim, Rezaul and Arefin, Mohammad Shamsul, Developing a Framework for Online Practice Examination and Automated Score Generation (January 14, 2019). International Journal of Computer Science & Information Technology (IJCSIT) Vol 10, No 6, December 2018, Available at SSRN: https://ssrn.com/abstract=3315086
5. Timakova, Y., &amp; Bakon, K. A. (2018). Bloom’s taxonomy-based Examination Question Paper Generation System. International Journal of Information System and Engineering, 6(2), 76–92. https://doi.org/10.24924/ijise/2018.11/v6.iss2/76.92
6. Wang, X., Wang, Z., Huang, W., Wen, G., &amp; Zhang, S. (2017). Design and realization of Test Question Bank Database System. AIP Conference Proceedings. https://doi.org/10.1063/1.4982561
7. M., P. A. R. T. H. A. S. A. R. A. T. H. Y., &amp; R., A. N. A. N. T. H. A. S. A. Y. A. N. A. M. (2016). Development and validation of web-based Question Bank and evaluation of its utility among students and teachers. i-Manager’s Journal of Educational Technology, 13(1), 35. https://doi.org/10.26634/jet.13.1.6016
8. Dickinson, J. R. (2015). A taxonomy assessment and item analysis of a retailing management multiple-choice Question Bank. Developments in Marketing Science: Proceedings of the Academy of Marketing Science, 329–330. https://doi.org/10.1007/978-3-319-10912-1_111

9. Gooi, A. C. C., &amp; Sommerfeld, C. S. (2014). Medical School 2.0: How we developed a student-generated question bank using small group learning. Medical Teacher, 37(10), 892–896. https://doi.org/10.3109/0142159x.2014.970624

10. Kale, V. M., &amp; Kiwelekar, A. W. (2013). An algorithm for question paper template generation in question Paper generation system. 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE). https://doi.org/10.1109/taeece.2013.6557281

11. Purohit, V. K., Kumar, A., Jabeen, A., Srivastava, S., Goudar, R. H., Shivanagowda, & Rao, S. (2012). Design of Adaptive Question Bank Development and Management System. 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing. https://doi.org/10.1109/pdgc.2012.6449828

12. Liske, H. (2011). A framework for automated generation of examination questions from web based semantically treated search results. Proceedings of the 12th International Conference on Computer Systems and Technologies - CompSysTech '11. https://doi.org/10.1145/2023607.2023693

13. Questionnaire generator: Web application using cloud computing. (2014). E-Commerce, E-Business and E-Service, 133–138. https://doi.org/10.1201/b17084-26

14. Bhatia, R., Gautam, V., & Garg, Y. K. (2019). Dynamic question answer generator: An enhanced approach to question generation. International Journal of Trend in Scientific Research and Development, Volume-3(Issue-4), 785–789. https://doi.org/10.31142/ijtsrd23730

15. MPhil, M. R., &amp; K, G. (2019). Automatic Question Paper Generator System. International Journal of Trend in Scientific Research and Development, Volume-3(Issue-3), 138–139. https://doi.org/10.31142/ijtsrd21646

16. Pranav Nair, M., Paul, J., Babu, A., &amp; Singh, M. (2020). Online question paper and Question Bank Generator and student portal. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 258–261. https://doi.org/10.32628/cseit206357

17. Baviskar, V. (2021). Question paper generator. International Journal for Research in Applied Science and Engineering Technology, 9(VI), 2076–2078. https://doi.org/10.22214/ijraset.2021.35283

18. Kiran, F., Gopal, H., &amp; Dalvi, A. (2017). Automatic Question Paper Generator System. International Journal of Computer Applications, 166(10), 42–47. https://doi.org/10.5120/ijca2017914138

19. AbuMansour, H. Y. (2017). Proposed bio-authentication system for Question Bank in Learning Management Systems. 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA). https://doi.org/10.1109/aiccsa.2017.215

20. Dwivedi, P., Shankar, R. T., Meghana, B., Sushaini, H., Sudeep, B. R., & Pooja, M. R. (2020). Designing an adaptive question bank and question paper generation management system. Advances in Intelligent Systems and Computing, 965–973. https://doi.org/10.1007/978-981-15-3514-7_72

21. Nalawade G, Ramesh, R (2016) Automatic generation of question paper from user entered specifications using a semantically tagged question repository. In: 18th IEEE eighth international conference on technology for education (T4E). 2–4 Dec 2016. https://doi.org/10.1109/t4e.2016.038

22. Zahorian SA, Lakdawala VK, Gonzalez OR, Starsman S, Leathrum Jr JF (2001) Question model for intelligent questioning systems in engineering education. In: 31st ASEE/IEEE frontiers in education conference. 10–13 Oct 2001. https://doi.org/10.1109/fie.2001.963871

23. Cen G, Dong Y, Gao W et al. (2010) A implementation of an automatic examination paper generation system. Math Comput Model 52: 1339–1342. https://doi.org/10.1016/j.mcm.2009.11.010

24. Franzke, M, Kintsch, E, Caccamise, D, et al. (2005) Summary Street®: Computer Support for Comprehension and Writing. J Educ Comput Res 33(1):53–80. https://doi.org/10.2190/DH8F-QJWM-J457-FQVB

25. Lemaire B, Dessus P (2001) A system to assess the semantic content of student essays. J Educ Comput Res 24:305–320. https://doi.org/10.2190/G649-0R9C-C021-P6X3

26. Dalton E (2018) The new bloom’s taxonomy, objectives, and assessments (December 3 2003). Retrieved from http://gaeacoop.org/dalton/publications/new_bloom.pdf on 2 Feb 2018

27. Dwivedi P, Rajgopal K, Srinivasan RK (2016) Multipurpose indian language evaluation system and question bank. India International Science Festival (IISF)—Young Scientists’ Conclave (YSC) 8–11 Dec 2016

28. Forehand M (2011) Bloom’s taxonomy-emerging perspectives on learning, teaching and technology. The University of Georgia
29. Anderson, J (2005) Mechanically inclined: building grammar, usage, and style into writer's workshop. Stenhouse Publishers
30. E-commerce web application by using Mern Technology. (2021). International Journal for Modern Trends in Science and Technology, 7(05), 1–5. https://doi.org/10.46501/ijmtst0705001
31. Saravanan Raju, S.Soundararajan, V.Loganathan. (2021). MERN Stack Web Application. Annals of the Romanian Society for Cell Biology, 25(6), 6325–6332. Retrieved from https://annalsofrscb.ro/index.php/journal/article/view/6683
32. Monika Mehra, Manish Kumar, Anjali Maurya, Charu Sharma, Shanu. (2021). MERN Stack Web Development. Annals of the Romanian Society for Cell Biology, 25(6), 11756–11761. Retrieved from https://www.annalsofrscb.ro/index.php/journal/article/view/7719
33. Deepika, N. M., Bala, M. M., &amp; Kumar, R. (2021). Design and implementation of intelligent virtual laboratory using RASA framework. Materials Today: Proceedings. https://doi.org/10.1016/j.matpr.2021.01.226
34. Bawane, M. (2022). A review on technologies used in mern stack. International Journal for Research in Applied Science and Engineering Technology, 10(1), 479–488. https://doi.org/10.22214/ijraset.2022.39868
35. Patil, D. R., Gentyal, V., Mudaliar, V., Kanpurne, G., & Ambi, D. (2022). College website using Mern Stack. International Journal for Research in Applied Science and Engineering Technology, 10(4), 1096–1098. https://doi.org/10.22214/ijraset.2022.41450
36. Kavade, P. (2019). Innovative recruitment techniques in job portal with Mern Stack. International Journal for Research in Applied Science and Engineering Technology, 7(10), 810–814. https://doi.org/10.22214/ijraset.2019.10122
37. Aneesh R, Ajmal Shah, Abhishek D M, Aishwarya S.R, & Thaseen Taj. (2020). Community web application for event management platform. International Journal of Progressive Research in Science and Engineering, 1(5), 116–120. Retrieved from https://journals.grdpublications.com/index.php/ijprse/article/view/165
38. Saundariya, K., Abirami, M., Senthil, K. R., Prabakaran, D., Srimathi, B., & Nagarajan, G. (2021). Webapp service for booking handyman using mongodb, express JS, react JS, node

JS. 2021 3rd International Conference on Signal Processing and Communication (ICPSC). https://doi.org/10.1109/icspc51351.2021.9451783

39. Biswas, N. (2021). Building a messaging app with mern. MERN Projects for Beginners, 95–168. https://doi.org/10.1007/978-1-4842-7138-4_4
40. Vhandale, A., Gandhak, S., Karhale, S., Prasad, S., &amp; Bachwani, P. S. (2022). Mern stack: Technologies used for web development. International Journal for Research in Applied Science and Engineering Technology, 10(2), 311–318. https://doi.org/10.22214/ijraset.2022.40247
41. Petralba, J. (2020). Wordnet semantic relations in a chatbot. Recoletos Multidisciplinary Research Journal, 8(2), 15–34. https://doi.org/10.32871/rmrj2008.02.02
42. Vasanthi, D., Sivasakthi, T., Abarna, V., &amp; Arthi, R. (2021). Design and development of car rentalwebsite using mern stack. 2021 International Conference on Computing, Communication and Green Engineering (CCGE). https://doi.org/10.1109/ccge50943.2021.9776473
43. Harjani, Mohak and Singh, Neetu and Behera, Priyanka, Sing It along Using Mern Stack (February 8, 2022). Available at SSRN: https://ssrn.com/abstract=4029349 or http://dx.doi.org/10.2139/ssrn.4029349
44. Porter, P., Yang, S., &amp; Xi, X. (2019). The design and implementation of a restful IOT service using the Mern Stack. 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW). https://doi.org/10.1109/massw.2019.00035
45. Design and application of University Students Management System based on web platform. (2016). Revista De La Facultad De Ingeniería. https://doi.org/10.21311/002.31.8.10