QUESTION PAPER MANAGEMENT SYSTEM

by

CHINTA LIKHITHA SOWMYA CHITTI LAKSHMI 421130

GUNDA SRI HARINI 421146

Under the guidance of

GIREESH YELLASIRI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

TADEPALLIGUDEM-534101, INDIA

APRIL 2024

QUESTION PAPER MANAGEMENT SYTEM

Thesis submitted to
National Institute of Technology Andhra Pradesh
for the award of the degree

of

Bachelor of Technology

by

CHINTA LIKHITHA SOWMYA CHITTI LAKSHMI 421130
GUNDA SRI HARINI 421146

Under the guidance of

GIREESH YELLASIRI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH
TADEPALLIGUDEM-534101, INDIA
APRIL 2024

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

CHINTA LIKITHA SOWMYA CHITTI LAKSHMI

421130

Date:_____

GUNDA SRI HARINI

421146

Date:_____

CERTIFICATE

It is certified that the work contained in the thesis titled "**QUESTION PAPER MANAGEMENT SYSTEM**" by "CHINTA LIKHITHA SOWMYA CHITTI LAKSHMI, bearing Roll No: 421130" and "GUNDA SRI HARINI,bearing Roll No: 421146" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature
GIREESH YELLASIRI
DCSE
N.I.T. Andhra Pradesh
April, 2024

Abstract

The goal of the Question Paper Management System is to make it easier for students to access, arrange, and handle their academic question papers. The system starts with an easy-to-use sign-up form that makes sure users enter all the necessary data, including name, email address, department, roll number, current year, and password. Certain restrictions are enforced by the form, including those regarding the roll number format, department selection, email domain, and options for the current year. Users are taken to the main page after completing the register process.

Users can bookmark question papers in the favorites section for future use, making it simple to access commonly used resources. Users have the option to save question papers to their favorites for quick access at a later time or download them straight away. The ease and effectiveness with which users can obtain pertinent academic resources is improved by this feature.

In general, the Question Paper Management System provides students with an all-inclusive way to effectively organize and retrieve academic question papers, improving their learning environment and output.

TABLE OF CONTENTS

Contents

1 Project Idea

Students sometimes struggle with the difficulty of effectively maintaining and accessing academic resources, such as question papers, in the hectic world of academia. These papers are difficult for students to find and properly arrange because they are dispersed over multiple platforms, despite the fact that they are crucial for exam preparation and academic progress. Seeing the need for a more efficient alternative, the Question Paper Management System shows up as a convenient and efficient answer.

The urgent need for a centralized archive for academic question papers led to the creation of the Question Paper Management System, a user-centric platform. This system, which was created primarily to streamline the process of managing, organizing, and gaining access to question papers, has the potential to completely transform the way students engage with their course materials.

The Question Paper Management System provides students with an easy-to-use interface that makes it simple to sign up and obtain the question papers they want. With features like personalized user profiles, extensive search capabilities, and bookmarking of favorite papers, the system makes it simple for students to take charge of their academic careers.

In-depth analysis of the QPMS's main functions, user interface, functionality, and possible effects on students' academic experiences are provided in this paper. Through an explanation of the system's architecture, design concepts, and user interface, this paper seeks to offer a thorough grasp of its importance in contemporary learning environments.

2 Requirements

The Question Paper Management System is a web application designed to streamline the process of managing and accessing question papers for various departments and courses within an educational institution. The system provides a user-friendly interface where students can register, authenticate, and access question papers based on their preferences and requirements. The application aims to improve efficiency and organization in the distribution and retrieval of question papers, making it easier for students to prepare for exams.

2.1 User Registration and Authentication:

1. Sign-up form for new users, capturing essential details like name, roll number, department, email, current year, and password.
2. Validation rules:
    - Roll number should be a 6-digit number
    - Department options: CSE, ECE, EEE, MECH, CIVIL, CHEM, MME, BIO
    - Email in the format rollno@student.nitandhra.ac.in
    - Current year options: I, II, III, IV
3. Sign-in functionality for existing users with email and password validation.
4. User profile page displaying user information with an option to edit personal details.

2.1.1 Question Paper Management:

- Home page with a list of department buttons for navigating to respective question papers.
- Department-wise categorization of question papers, allowing users to view question papers specific to their department.
- Comprehensive search and filter options based on year, subject code, subject name, and exam type (minor-1, minor-2, mid, semester).
- Sort options for question papers based on academic year, enabling users to organize question papers chronologically.
- ”All” option for selecting all departments, years, semesters, subjects, and exam types, providing a comprehensive view of available question papers.
- Download functionality for question papers, allowing users to save and access them offline.
- Favorites section in the home page, enabling users to bookmark frequently used question papers for easy access.

3 ER Diagram:

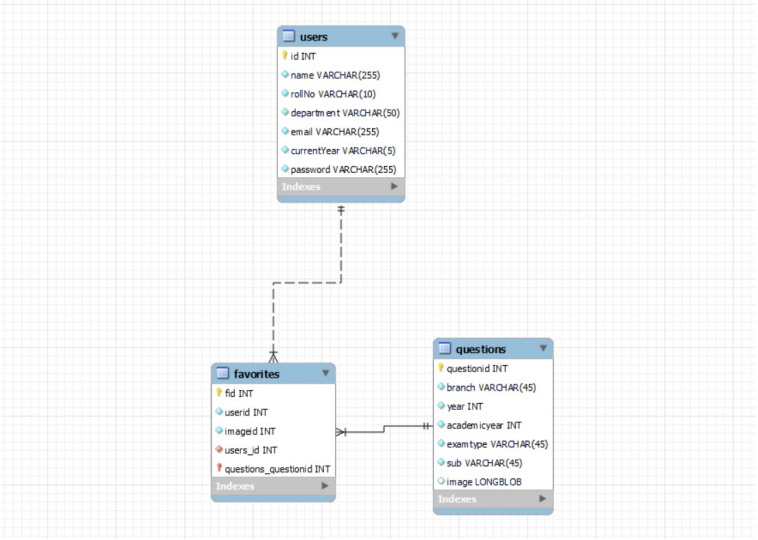

Figure 1: The ER Diagram

4 Architecture:

- Client-server architecture with a clear separation of concerns between the front-end (client-side) and back-end (server-side) components.
- Client-side Architecture:
  - Front-end built using React, a JavaScript library for building user interfaces.
  - React components organized following a modular approach.
  - React Router for client-side routing .
  - Axios library for making HTTP requests to the server-side APIs.
  - Unidirectional data flow pattern, with data passed down from parent to child components.
  - React hooks for managing state and lifecycle methods.
  - Responsive and mobile-friendly user interface, leveraging CSS frameworks like Bootstrap or Material-UI.
- Server-side Architecture:
  - Back-end built using Node.js and Express.js.
  - RESTful API architecture, exposing endpoints for various operations.
  - Server-side logic organized into separate modules or controllers.
  - Middleware functions for tasks like request parsing, authentication, and error handling.
  - MySQL database for storing user information, question paper metadata, and other relevant data.
  - Possible use of an Object-Relational Mapping (ORM) library like Sequelize or Prisma.
  - Security measures like input validation, sanitization, and encryption.
- Data Flow and Communication:
  - Client-side components dispatch actions or make API calls to the server-side.
  - Server-side processes requests, performs operations, and sends back response data.
  - Client-side receives response data and updates the user interface accordingly.
  - File downloads (e.g., question paper PDFs) served as file streams by the server-side.

5 Design:

5.1 User Interface Design

- Clean and modern user interface (UI) design following the latest UI/UX best practices.
- Consistent layout and color scheme throughout the application for a cohesive experience.
- Responsive design using CSS frameworks like Bootstrap or Material-UI to ensure optimal viewing experience across various devices and screen sizes.
- Intuitive navigation with clear labels, icons, and menu structures for easy navigation.
- Appropriate use of typography, spacing, and visual hierarchy to guide the user's focus and improve readability.
- Effective use of animations and transitions for smooth user interactions and feedback.

5.2 Component-based Architecture

- Modular design approach with the application divided into reusable and self-contained components.
- Separate components for user management (sign-up, sign-in, profile), question paper management (search, filter, sort, download), and other functional areas.
- Efficient component communication and data flow using React's props and state management mechanisms.
- Reusable components for common UI elements like buttons, forms, modals, and alerts.

5.3 Routing and Navigation

- Client-side routing using React Router for seamless navigation between different views and components.
- Clear and intuitive URL structure for easy bookmarking and sharing of specific application states.
- Handling of different routes and components based on user authentication status.

5.4 State Management

- Effective use of React hooks (e.g., useState, useEffect) for managing component state and lifecycle methods.
- Centralized state management using a state management library like Redux or MobX for complex applications with shared state across multiple components.
- Proper handling of async operations and side effects using hooks like useEffect or utilities like React Suspense.

5.5 Form Handling and Validation

- Efficient form handling and validation using library like Formik or React Hook Form.
- Client-side validation for user input fields with appropriate error messages and feedback.
- Integration with server-side validation for enhanced security and data integrity.

5.6 Error Handling and Feedback

- Proper error handling and user feedback for various scenarios like network errors, server errors, or invalid user input.
- Clear and user-friendly error messages and notifications to guide the user.
- Appropriate use of loading indicators and progress bars for long-running operations.

5.7 Security and Performance

- Implementation of security best practices like input validation, sanitization, and encryption (both client-side and server-side).
- Adherence to secure coding guidelines and security principles like the principle of least privilege.
- Optimization techniques like code splitting, lazy loading, and memoization for improved performance and faster load times.
- Proper handling of async operations and side effects to prevent performance bottlenecks.

5.8 Accessibility

- Adherence to accessibility guidelines and best practices (e.g., WCAG, Section 508) to ensure the application is usable by individuals with disabilities.
- Appropriate use of semantic HTML elements, ARIA roles, and keyboard navigation support.
- Proper contrast ratios, alternative text for images, and other accessibility considerations.

5.9 Testing and Quality Assurance

- Implementation of unit tests, integration tests, and end-to-end tests for different components and functionality.
- Continuous integration and deployment pipelines for automated testing and quality assurance.
- Code reviews and static code analysis for maintaining code quality and adhering to best practices.

article graphicx caption

6 Execution Screenshots

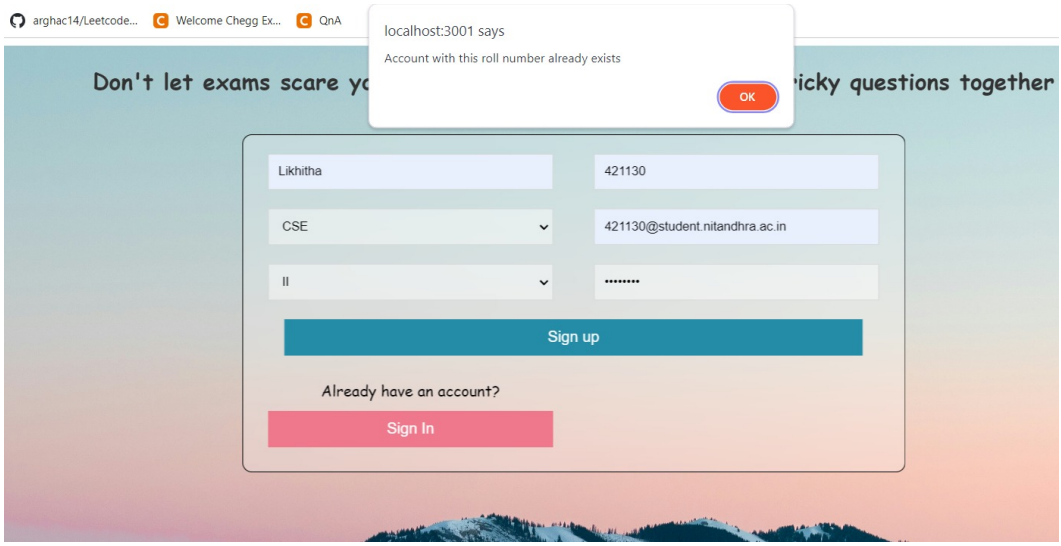

Figure 2: Checking if the Rollno is same in Rollno and EmailId Column

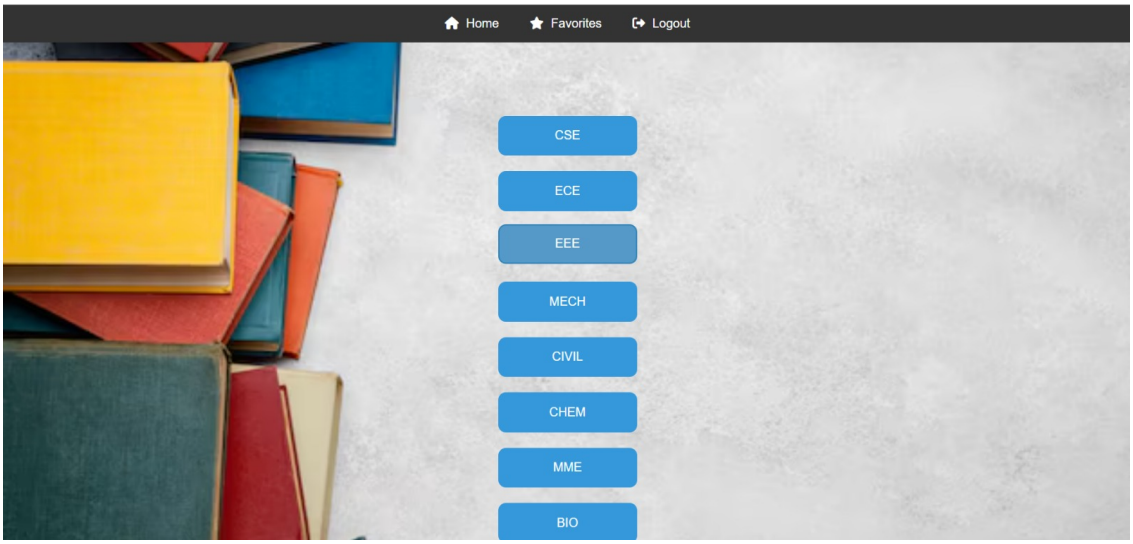

Figure 3: Homepage with all the Departments

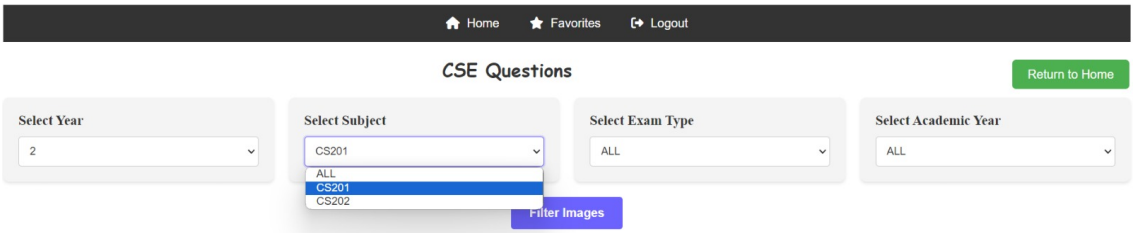

Figure 4: Courses available in 2nd year

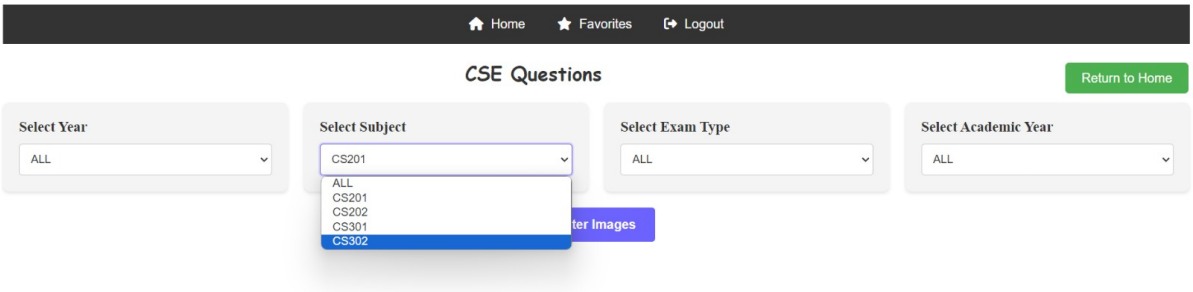

Figure 5: Courses available for CSE students

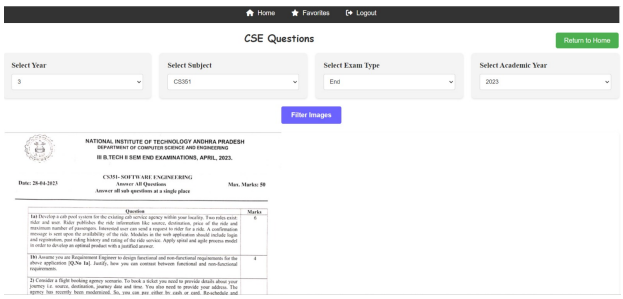

Figure 6: The Question paper based on the input

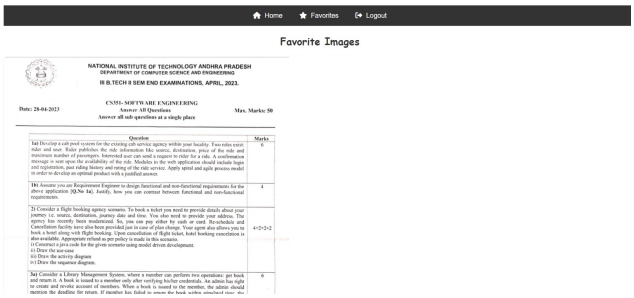

Figure 7: Adding Question Paper into Favourites

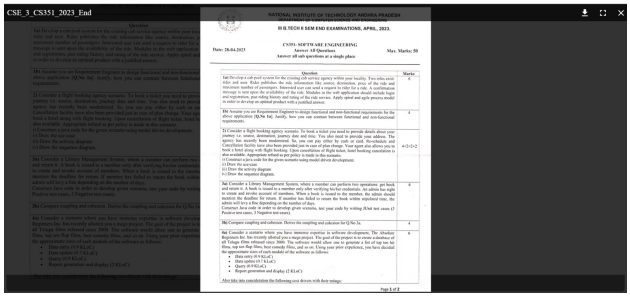

Figure 8: Download the question Paper

7 Conclusion

In conclusion, our ”Question Paper Management System” addresses the need for a centralized platform for accessing and managing academic question papers. The system's user-friendly design, robust search and filter capabilities, and future scalability make it a valuable tool for students seeking organized access to their academic materials.

While the current version focuses on question papers, future iterations will incorporate reference books to further enhance the system's utility. The feedback and suggestions from users will drive continuous improvement and refinement of the system, ensuring its relevance and effectiveness in meeting the academic needs of students.

8 Future Work

In conclusion, our Question Paper Management System is a valuable tool for students, offering a convenient and organized way to access academic question papers. The current implementation focuses on question papers, providing features like user registration, authentication, navigation to question papers by department, filtering and sorting options, and favorite management.

Future work will expand the system to include a References section, enhancing its utility by incorporating a repository for reference materials. This addition will further support students in their academic endeavors by providing a centralized platform for accessing both question papers and reference resources. The continuous development and improvement of our system aim to meet the evolving needs of students and facilitate their academic success.

9 Assessment

The Question Paper Management System is a web application designed to streamline the process of managing and accessing question papers for various departments and courses within an educational institution. The system provides a user-friendly interface where students can register, authenticate, and access question papers based on their preferences and requirements. The application aims to improve efficiency and organization in the distribution and retrieval of question papers, making it easier for students to prepare for exams.

9.1 User Registration and Authentication:

1. Sign-up form for new users, capturing essential details like name, roll number, department, email, current year, and password.
2. Validation rules:
   - Roll number should be a 6-digit number
   - Department options: CSE, ECE, EEE, MECH, CIVIL, CHEM, MME, BIO
   - Email in the format rollno@student.nitandhra.ac.in
   - Current year options: I, II, III, IV
3. Sign-in functionality for existing users with email and password validation.
4. User profile page displaying user information with an option to edit personal details.

9.2 Question Paper Management:

- Home page with a list of department buttons for navigating to respective question papers.
- Department-wise categorization of question papers, allowing users to view question papers specific to their department.
- Comprehensive search and filter options based on year, subject code, subject name, and exam type (minor-1, minor-2, mid, semester).
- Sort options for question papers based on academic year, enabling users to organize question papers chrono- logically.
- ”All” option for selecting all departments, years, semesters, subjects, and exam types, providing a comprehensive view of available question papers.
- Download functionality for question papers, allowing users to save and access them offline.
- Favorites section in the home page, enabling users to bookmark frequently used question papers for easy access.

10 References

1. React Documentation: https://reactjs.org/docs/getting-started.html
2. Node.js Documentation: https://nodejs.org/en/docs/
3. Express.js Documentation: https://expressjs.com/en/4x/api.html
4. MySQL Documentation: https://dev.mysql.com/doc/
5. Sequelize Documentation: https://sequelize.org/master/
6. Prisma Documentation: https://www.prisma.io/docs/
7. Axios Documentation: https://axios-http.com/docs/intro
8. Bootstrap Documentation: https://getbootstrap.com/docs/5.0/getting-started/introduction/
9. Material-UI Documentation: https://mui.com/getting-started/usage/