

PYQ Portal: A Full-Stack Web Platform for Managing Previous Year Question Papers and Academic Resources

Akash Golhani

*Dept. of Information Technology
Barkatullah University Institute of
Technology
Bhopal (M.P), India
akashgolhani083@gmail.com*

Sagar Chaurasiya

*Dept. of Information Technology
Barkatullah University Institute of
Technology
Bhopal (M.P), India
chaurasiyasagar1000@gmail.com*

Sagar Rahangdale

*Dept. of Information Technology
Barkatullah University Institute of
Technology
Bhopal (M.P), India
Sagarrahangdale123@gmail.com*

Shivendra Kamthare

*Dept. of Information Technology
Barkatullah University Institute of
Technology
Bhopal (M.P), India
shivendrakamthare@gmail.com*

Mrs.Nishtha Chouhan

*Asst. Prof. at Dept. of Information
Technology
Barkatullah University Institute of
Technology
Bhopal (M.P), India*

Mr.Pawan Singh Rajput

*Asst. Prof. at Dept. of Information
Technology
Barkatullah University Institute of
Technology
Bhopal (M.P), India*

Dr.Rachna Kulhare

*Asst. Prof. at Dept. Information
Technology
Barkatullah University Institute of
Technology
Bhopal (M.P), India*

Abstract: The rapid growth of digital education has increased the need for efficient platforms that provide students with quick access to academic resources. This research presents the design and development of a web-based Previous Year Question Paper (PYQ) Management System built using the MERN stack (MongoDB, Express.js, React/Node.js). The platform is designed to simplify the process of accessing, managing, and sharing academic materials such as question papers, solutions, syllabi, and study resources.

The system provides a structured hierarchy for universities, branches, semesters, and subjects, enabling students to easily navigate and download relevant documents. In addition, the platform includes features such as an admin dashboard for content management, cloud-based file storage, live exam updates, and a student discussion section for communication and collaboration. The integration of secure authentication

and responsive design ensures both usability and accessibility across devices.

The proposed system aims to reduce the difficulty students face in collecting academic resources from multiple sources while improving organization and availability through a centralized digital platform. The project demonstrates the practical implementation of full-stack web technologies in the education domain and highlights how modern web applications can enhance academic resource management and student engagement.

Keywords: Previous Year Question Papers, MERN Stack, Web Application, Academic Resource Management, Student Portal, Cloud Storage, Full-Stack Development, Educational Platform, MongoDB, Express.js, Node.js.



Fig 1: Home Page

1.Introduction

In the modern education system, students often face difficulties in accessing previous year question papers, syllabi, solutions, and other academic resources from different sources. Traditional methods of sharing study materials are time-consuming, unorganized, and inefficient. To solve this problem, a centralized digital platform is required where students can easily access and manage academic resources in one place.

This project presents a web-based Previous Year Question Paper (PYQ) Management System developed using the MERN stack, including MongoDB, Express.js, React.js, and Node.js. The platform allows students to browse universities, branches, semesters, and subjects in a structured manner and access related PDFs, solutions, and study materials. It also includes features such as live exam updates, a student discussion section, cloud-based file storage, and an admin dashboard for managing content efficiently.

The main objective of the system is to improve accessibility, organization, and availability of academic resources while providing a user-friendly and scalable educational platform. The project demonstrates how modern full-stack web technologies can be effectively used to enhance academic resource management and student learning experiences.

1.1 Objective

The main objective of this project is to develop a centralized and user-friendly web platform that helps students easily access and manage academic resources such as previous year question papers, solutions, syllabi, and study materials. The system is designed to improve resource organization, accessibility, and

communication among students while reducing the dependency on scattered and offline sources of information.

- Centralized Resource Management:** To create a single platform where students can access all academic resources including PYQs, solutions, syllabi, and PDFs in an organized manner.
- Structured Academic Navigation:** To provide a hierarchical system based on universities, branches, semesters, and subjects for easy browsing and resource discovery.
- Efficient Admin Control:** To implement an admin dashboard that allows secure uploading, updating, and management of study materials and academic content.
- Improved Accessibility:** To ensure students can access resources anytime and from any device through a responsive and cloud-based web application.
- Student Interaction and Updates:** To include features such as live exam updates and discussion sections that encourage communication and information sharing among students.
- Scalable Full-Stack Development:** To build a scalable and efficient application using MERN stack technologies for better performance, maintainability, and future enhancements.

1.1 Research Background

The rapid growth of digital learning platforms has transformed the way students access academic resources. Traditional methods of collecting previous year question papers (PYQs), syllabus documents, and study materials are often unorganized and time-consuming. Students usually depend on seniors, offline notes, or scattered online sources, which creates difficulty in finding reliable and updated content. To overcome these issues, educational institutions and developers are increasingly adopting web-based academic management systems.

Need for Centralized Academic Resources

Students require a single platform where they can easily access PYQs, syllabus PDFs, solutions, and important academic updates. A centralized system improves

accessibility, reduces manual effort, and saves time for both students and administrators.

Role of Full Stack Web Development

Modern full stack technologies such as MongoDB, Express.js, Node.js, and EJS/Bootstrap enable the development of scalable and responsive educational platforms. These technologies support secure data handling, dynamic content rendering, and efficient database management.

Importance of Cloud-Based File Management

Managing educational PDFs and documents requires reliable storage solutions. Cloudinary and MongoDB Atlas provide cloud-based storage and database services that help in securely storing and retrieving files with high availability and performance.

1.2 Research Motivation

The main motivation behind this research is to solve the difficulties students face while searching for previous year question papers, syllabus PDFs, and study resources from multiple unorganized sources. Many students spend significant time collecting academic materials, which affects their preparation and productivity.

The proposed PYQ's Portal aims to provide a centralized, user-friendly, and efficient platform where students can easily access study materials, receive live academic updates, and interact with other students. The project also focuses on simplifying content management through an admin dashboard and improving accessibility using modern full stack web technologies and cloud-based storage solutions.

2. PROPOSED SYSTEM

The proposed system is a full-stack web application designed to provide students with an organized and centralized platform for accessing Previous Year Question Papers (PYQs), syllabus PDFs, solutions, and other academic resources. The system simplifies the process of searching and managing study materials by

categorizing content according to university, branch, semester, and subject.

The platform includes both student and admin modules. Students can browse and download study resources, receive live academic updates, and interact with other users through the comment section. The admin panel allows administrators to upload, manage, and organize all academic content efficiently. Cloud-based storage integration ensures secure handling of PDFs and images, while the responsive user interface improves accessibility across devices.

Key Components of the Proposed System

Student Portal: The student portal is designed to provide quick and easy access to academic resources in an organized manner. Students can navigate through universities, branches, semesters, and subjects to find relevant question papers, syllabus PDFs, and solutions. The portal also includes search and filtering functionality to improve accessibility and reduce the time required to find study materials. Its responsive design ensures smooth usage on both desktop and mobile devices.

Admin Dashboard: The admin dashboard enables administrators to efficiently manage the entire platform from a centralized interface. Admins can add, edit, and delete universities, subjects, semesters, and uploaded resources without requiring technical knowledge. The dashboard also supports secure file uploads and content management, ensuring that the academic data remains updated and properly organized. This reduces manual effort and improves system maintainability.

Resource Management System: The resource management system organizes all academic materials in a structured hierarchy to simplify data handling and retrieval. Question papers, solutions, syllabus documents, and other study resources are categorized systematically according to educational structure. This organized approach improves navigation efficiency and helps students quickly locate the required files. The system also ensures scalability for adding new subjects and universities in the future.

Cloud Storage Integration: The proposed system integrates Cloudinary for secure cloud-based storage of PDFs, images, and other uploaded resources. This integration reduces server load and improves file

accessibility and reliability. Cloud storage also ensures that uploaded files remain secure and easily manageable while supporting faster content delivery. The feature enhances overall system performance and storage efficiency.

3. METHODOLOGY

The methodology of the proposed PYQ Portal system focuses on designing and developing a structured web platform that simplifies access to academic resources for students. The system follows a full-stack development approach using the MERN stack and cloud-based services for efficient data management and scalability. The development process includes requirement analysis, database design, backend development, frontend implementation, cloud integration, and deployment.

Requirement Analysis

The first step of the methodology involves analyzing the problems faced by students while searching for previous year question papers and study materials. Requirements such as easy navigation, categorized resources, secure admin management, and responsive design were identified. The system was planned to provide both student-side accessibility and admin-side content management features.

System Design

The system architecture was designed to maintain a clear hierarchy of universities, branches, semesters, and subjects. Database models were created using MongoDB and Mongoose to store academic resources efficiently. User interface layouts and workflow diagrams were planned to ensure smooth navigation and better user experience across all devices.

Frontend Development

The frontend of the application was developed using EJS, HTML, CSS, Bootstrap, and JavaScript. Responsive web pages were created to allow students to browse and access study materials easily. Interactive components such as search functionality, navigation

menus, and dynamic content rendering were implemented to improve usability and accessibility.

Backend Development

The backend was developed using Node.js and Express.js to handle server-side operations and API requests. RESTful APIs were implemented for managing universities, subjects, semesters, and uploaded files. Authentication and authorization mechanisms were added to ensure secure admin access and proper content management.

Database and Cloud Integration

MongoDB Atlas was used as the cloud database to store structured academic data. Cloudinary integration was implemented for storing PDFs and images securely in the cloud. Multer was used for handling file uploads efficiently before transferring them to cloud storage. This approach improved scalability and reduced local server storage dependency.

Testing and Deployment

The system was tested for functionality, responsiveness, and data handling to ensure reliable performance. Features such as file uploads, search operations, and admin controls were verified to remove errors and improve stability. Finally, the application was deployed on Render, allowing users to access the platform online with cloud-hosted database support through MongoDB Atlas.

4. TECHNOLOGY STACK

The proposed PYQ Portal system is developed using modern web technologies to ensure scalability, responsiveness, and efficient data management. The technology stack includes frontend technologies for user interaction, backend technologies for server-side processing, database services for data storage, and cloud platforms for deployment and file management.

Frontend Technologies

The frontend of the system is developed using **HTML5, CSS3, Bootstrap, EJS, and JavaScript**. These technologies are used to create a responsive and interactive user interface that allows students to

navigate through universities, semesters, and subjects efficiently. Bootstrap helps in building a mobile-friendly design, while EJS enables dynamic rendering of web pages.

Backend Technologies

The backend is developed using **Node.js** and **Express.js**. Node.js provides a fast and scalable runtime environment, while Express.js is used to build RESTful APIs and manage server-side logic. These technologies handle authentication, data processing, file uploads, and communication between the frontend and database.

Database Management

The system uses **MongoDB Atlas** as the cloud-based NoSQL database. MongoDB stores information related to universities, subjects, users, PDFs, and announcements in a structured format. **Mongoose ORM** is used to define schemas and simplify database operations.

Cloud Storage and File Handling

Cloudinary is integrated for secure storage and management of PDF files and images uploaded through the admin panel. **Multer** is used as middleware for handling file uploads before transferring them to cloud storage efficiently.

Deployment and Hosting

The application is deployed on **Render**, which provides cloud hosting for the backend and frontend services. MongoDB Atlas is used for remote database hosting, enabling reliable access and scalability for users from different locations.

Development Tools

Various development tools such as **Visual Studio Code**, **Git**, and **GitHub** are used for coding, version control, and project management. These tools help in maintaining code quality, collaboration, and efficient software development practices.

5. FEATURES

The PYQ Portal provides several useful features that help students access study materials easily and improve the overall learning experience. The system is designed to be user-friendly, secure, and efficient for both students and administrators.

Previous Year Question Paper Access: The platform allows students to access and download previous year question papers based on university, branch, semester, and subject. This helps students prepare effectively for examinations using organized academic resources.

Subject-wise Study Material: Students can view important study resources such as syllabus PDFs, solutions, expected questions, and subject notes in a structured manner. This centralized access reduces the time spent searching for academic content.

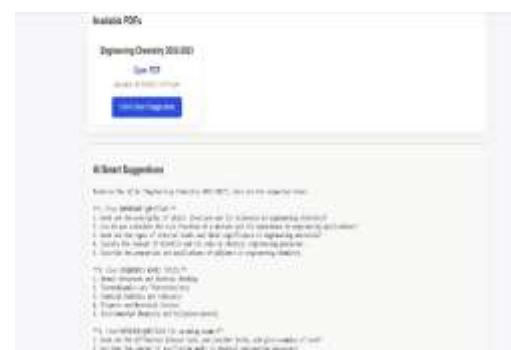


Fig: 2 PYQ PDF

Admin Management System: The system includes a secure admin panel through which administrators can add, update, or delete universities, subjects, semesters, and study materials. This ensures proper content management and regular updates.

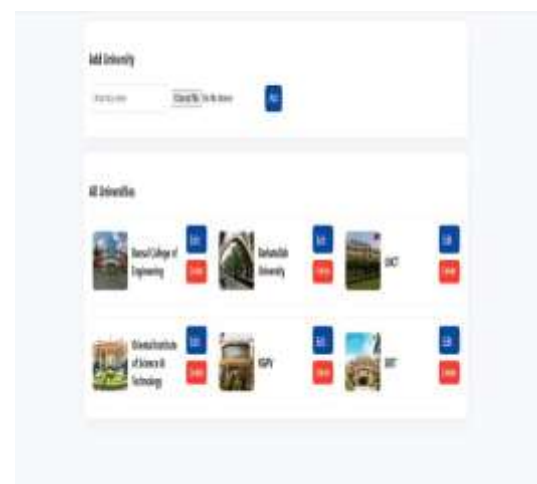


Fig: 3 Admin Management System

Live Update Section: A dedicated live update section is provided to share important notifications related to examinations, admissions, results, and academic announcements. This helps students stay informed about the latest updates.



Fig: 4 Live Updates

Student Interaction and Comment System: The platform includes a comment section where students can communicate, discuss doubts, and share information with each other. This feature improves collaboration and engagement among users.

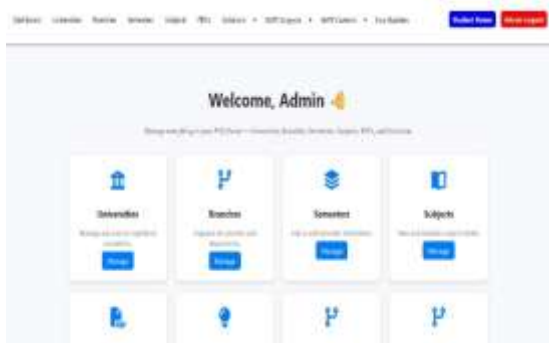


Fig: 5 Admin Panel

RESULT

The developed PYQ Portal successfully provides a centralized platform for accessing previous year question papers, study materials, and academic resources. The system improves the learning experience by organizing content based on university, branch, semester, and subject, making navigation simple and efficient for students.

The implementation of features such as PDF management, live updates, comment sections, and cloud-based storage enhanced the usability and functionality of the platform. Students were able to quickly search, view, and download study materials without facing complexity or delays.

The admin panel effectively managed uploads and content organization, reducing manual effort and ensuring that academic resources remain updated. The

responsive design also allowed smooth access across desktop and mobile devices.

Overall, the proposed system achieved its objective of creating an efficient, scalable, and user-friendly academic resource portal that simplifies exam preparation and improves accessibility to educational content.



Fig: 6 Subjects

7. FUTURE SCOPE OF PYQ PORTAL

The project can be further enhanced by adding advanced features and improvements to increase usability and performance. Future developments may include:

1. AI-Based Recommendation System

Artificial Intelligence can be integrated to recommend important questions, study materials, and personalized learning resources based on student activity.

2. Mobile Application Development

A dedicated Android and iOS mobile application can be developed to provide easier and faster access for students on smartphones.

3. Online Mock Tests

The system can include mock tests and quiz modules to help students practice and evaluate their preparation level.

4. Real-Time Notifications

Push notifications and email alerts can be added for important academic updates, exam schedules, and newly uploaded materials.

5. Advanced Search and Filtering

More advanced filtering and keyword-based search features can improve resource discovery and user experience.

6. Multi-University Expansion

The platform can be expanded to support multiple universities and competitive examination resources on a larger scale.

8. CONCLUSION

The PYQ Portal was successfully developed as a full-stack web application to simplify access to previous year question papers, study materials, and academic resources for students. The system provides a structured and user-friendly platform where users can easily browse subjects, download PDFs, receive live updates, and interact through the comment section.

The integration of technologies such as MongoDB, Express.js, Node.js, Cloudinary, and Bootstrap helped in building a secure, responsive, and scalable application. The admin panel also improved content management by allowing efficient uploading and organization of study materials.

Overall, the proposed system achieved its goal of improving academic resource accessibility and providing students with a centralized platform for better exam preparation and learning support.

9. REFERENCES

- MongoDB Documentation, “MongoDB Manual,” Available: <https://www.mongodb.com/docs/>
- Express.js Documentation, “Express – Node.js Web Framework,” Available: <https://expressjs.com/>
- Node.js Documentation, “Node.js Official Documentation,” Available: <https://nodejs.org/en/docs/>
- React.js Documentation, “React Official Documentation,” Available: <https://react.dev/>
- Bootstrap Documentation, “Bootstrap Frontend Framework,” Available: <https://getbootstrap.com/docs/>
- Cloudinary Documentation, “Cloudinary Media Management,” Available: <https://cloudinary.com/documentation>
- Mongoose Documentation, “Mongoose ODM for MongoDB,” Available: <https://mongoosejs.com/docs/>
- MDN Web Docs, “JavaScript Documentation,” Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- W3Schools, “HTML, CSS, JavaScript Tutorials,” Available: <https://www.w3schools.com/>
- Render Documentation, “Deploying Web Applications on Render,” Available: <https://render.com/docs>
- MongoDB Atlas Documentation, “Cloud Database Services,” Available: <https://www.mongodb.com/atlas/database>
- Multer Documentation, “Node.js Middleware for File Uploads,” Available: <https://github.com/expressjs/multer>
- GeeksforGeeks, “Web Development Tutorials and Concepts,” Available: <https://www.geeksforgeeks.org/>
- TutorialsPoint, “Web Technology Tutorials,” Available: <https://www.tutorialspoint.com/>
- freeCodeCamp, “Full Stack Development Resources,” Available: <https://www.freecodecamp.org/>
- IEEE Xplore Digital Library, “Research Papers on Educational Portals,” Available: <https://ieeexplore.ieee.org/>
- ResearchGate, “Research Articles on E-Learning Systems,” Available: <https://www.researchgate.net/>
- Journal of Educational Technology Systems, “Online Learning and Resource Management Systems,” Available through academic journals.
- Oracle Documentation, “Database and Backend Concepts,” Available: <https://docs.oracle.com/>
- Google Developers Documentation, “Web Development Best Practices,” Available: <https://developers.google.com/>

☐ CSS Tricks, “Frontend UI and Responsive Design Concepts,” Available: <https://css-tricks.com/>

☐ Stack Overflow Developer Community, “Programming Discussions and Solutions,” Available: <https://stackoverflow.com/>