Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Title of the Project

Web-Based School Administration System.

Abstract of the project

The **Web-Based School Administration System** is an integrated online platform designed to simplify and automate various administrative processes in schools. This system serves as a centralized hub where administrators, teachers, students, and parents can access relevant information, manage records, and communicate effectively. The platform offers features such as student registration, grade management, attendance tracking, timetable scheduling, and real-time notifications. With a user-friendly interface, secure login, and advanced data management tools, this system aims to streamline administrative tasks, reduce paperwork, and improve overall efficiency in school management.

Keywords

Generic Keywords:

Integration, Databases, Middleware, Programming

Specific Technology Keywords:

HTML, CSS, JavaScript, ReactJs, MongoDb, NodeJs

Project Type Keywords:

UI/UX Interface, API Integration, Testing, Security

Functional components of the project

Users of the system:

- School Administration
- Teacher
- Student
- Parents
- School Staff

Functionality:

A grains trading website typically consists of several functional components to provide a user-friendly and informative experience for visitors. Here are some essential functional components for a grains trading website:

- **Student Registration and Profile Management:**
  Allows administrators and teachers to register new students, update student profiles, and manage personal information such as contact details, emergency contacts, and more.

- **Grade Management:**
  Teachers can input and update grades, track student performance over time, and generate grade reports. Students and parents can access real-time grade information and performance analytics.

- **Class Timetable Management:**
  Administrators can create and manage class schedules, assign teachers to specific classes, and make adjustments as needed. Students and teachers can view their schedules in real-time.

- **Communication Tools:**
  Provides a messaging system where teachers can communicate with students and parents, and administrators can send out important announcements and notifications.

- **School Events and Announcements:**
  Enables administrators to post school events, holidays, and announcements, and notify all users with updates via email or app notifications.

- **Reporting and Analytics:**
  Generates comprehensive reports on student performance, attendance trends, class performance, and other key metrics, helping administrators make informed decisions.

- **Security and User Authentication:**
  Ensures that sensitive student and staff data is protected with strong user authentication methods, such as multi-factor authentication (MFA), and proper data encryption.

- **Role-Based Access Control (RBAC):**
  Provides different access levels for administrators, teachers, students, and parents, ensuring that each user only sees the information relevant to their role.

Non Functional Requirements:

- **Performance**: The website should load quickly and handle a high volume of concurrent users.
- **Scalability:** The system should be scalable to accommodate future growth.
- **Reliability**: Minimize downtime and ensure data integrity
- **Security**: Protect user data, transactions, and the website against security threats.
- **Usability**: Provide an intuitive and user-friendly interface.
- **Accessibility:** Ensure the website is accessible to users with disabilities.
- **Compatibility**: Support major web browsers and devices.
- **Compliance**: Comply with relevant legal and industry regulations.

Steps to start-off the project:

Creating a Grain Trading Website involves several steps and considerations to ensure its functionality, security, and user-friendliness. Here's an outline of the website making process:

1. Market Research and Planning

- Conduct market research to understand the needs and challenges of traders and farmers.
- Define the scope and objectives of the website.
- Identify the target audience and potential competitors.

2. Domain Name and Hosting

- Choose a suitable domain name that reflects the website's purpose.
- Select a reliable web hosting service to host the website.

3. Website Platform and Technology

- Decide on the technology stack for website development (e.g., HTML, CSS, JavaScript, backend language, and database).
- Choose whether to build the website from scratch or use a content management system (CMS).

4. Website Design

- Create a visually appealing and user-friendly design for the website.
- Ensure that the design is responsive and accessible on various devices (desktops, tablets, smartphones).

5. Frontend Development

- Implement the website's user interface based on the chosen design.
- Develop interactive features such as search options, filters, and user registration forms.
- Optimize the frontend for a smooth user experience.

6. Backend Development

- Set up the server and backend infrastructure.
- Implement the database to store user data, grain information, and transaction details using Node.js and Express.js.

7. Market Insights and Real-Time Data

- Integrate APIs or services to provide real-time market insights and grain prices.
- Display relevant market trends and data to users.

8. User Authentication and Security

- Implement a secure user authentication system.
- Ensure the website follows best practices for data security and protection.

9. Payment Gateway Integration

- Integrate a secure payment gateway to facilitate financial transactions.
- Enable various payment options to accommodate user preferences.

10. Logistics and Shipping Support

- Offer a system to coordinate logistics and shipping for grain delivery.
- Provide options for users to manage and track their shipments.

11. Testing and Quality Assurance

- Conduct thorough testing to identify and fix any bugs or issues.
- Test the website on different devices and browsers to ensure compatibility.

12. Launch and Marketing

- Launch the Grain Trading Website and make it accessible to users.
- Implement marketing strategies to attract users and promote the platform.
- Use social media, search engine optimization (SEO), and other marketing channels to reach the target audience.

13. Customer Support

- Set up customer support channels to assist users with inquiries and issues.
- Respond promptly to user feedback and continuously improve the platform based on user needs.

14. Monitor and Improve

- Regularly monitor website performance and user behaviour.
- Gather user feedback to identify areas for improvement and future feature enhancements.

Requirements

Hardware requirements

Number	Description	Alternatives (If available)
1	PC with 5 GB hard-disk ar MB RAM	Not-Applicable

Software requirements

Number	Description	Alternatives (If available)
1	Windows 10 or Window 11	Not Applicable
2	Visual Code Studio	CodePen
3	MongoDB	Redis
4	Windows	Linux
5	Postman	ThunderClient

Manpower requirements

2 students can complete this in 4 – 6 months if they work part-time on it.

Milestones and Timelines

No.	Milestone Name	Milestone Description	Timeline (in weeks)	Remarks
1	Requirements Specification	Complete specification document detailing the system (with appropriate assumptions). A presentation should also be prepared based on this.	2–3	Attempt should be made to add some more relevant functionalities other than those listed in the document.
2	Technology Familiarization	Understanding the technology needed to implement the project.	4–5	The presentation should be from the point of view of being able to apply it to the project, rather than from a purely theoretical perspective.
3	Database Creation	A database of at least 50 user entries and product entries should be created.	5–7	It is important to finalize the database at this stage itself so that development and testing can proceed with actual data.
4	High-Level & Detailed Design	Listing down all possible scenarios and then coming up with flowcharts and pseudocode to handle each scenario.	7–9	The scenarios should map to the requirement specification. For each requirement specified, a corresponding scenario should be there.

5	Implementation of Front-End	Implementation of the screens, including login screen and screens for each functionality.	10–12	During this milestone period, it would be a good idea for the team (or one member) to start working on a test plan for the entire system. This test plan should be updated over time.
6	Integrating Front-End with Database	The front-end developed in earlier milestones should be able to update and retrieve data from the database. The system should be ready for integration testing.	12–13	Integrating the front-end with the database is crucial. It connects the user interface (front-end) to the data storage and retrieval system (database).
7	Integration Testing	The system should be thoroughly tested by running all the test cases written for the system (from milestone 5).	14–15	Another 2 weeks should be allocated to handle any issues found during testing. After this, the final demo can be arranged.
8	Final Review	Issues found during the previous milestone are resolved, and the system is ready for the final review.	16–18	During the final review of the project, it should be checked that all the requirements specified in the initial stages are fulfilled.

Guidelines and References

- https://www.mongodb.com/
- https://react.dev/
- https://www.w3schools.com