BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Stay Ahead

BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Autonomous Institution, Accredited by NAAC With 'A' Grade

Name : Saranya T

Roll No. : 7376222AL195

Seat No. : 22

Project ID : 22

Module Name : Wiki Page Generation

Domain : Office Academics

Stack Allocation : MEAN Stack

---

1. INTRODUCTION:

1.1. Problem Statement:

Educators and administrators often encounter challenges when creating course pages in wiki page. Manually generating HTML code for course details such as the course title, faculty information, course objectives, and lesson plans can be time-consuming and error-prone. Additionally, ensuring consistency and accuracy across multiple course pages can be a daunting task. This website should allow administrators and faculty members to input course details in a structured manner and automatically generate well-formatted HTML code that can be easily integrated into a wiki page.

Primary Concerns:

1. **Manual HTML Code Generation:** Faculty members and administrators face the challenge of manually generating HTML code for course pages wiki page.

2. **Time-consuming Process:** Creating HTML code for course details is time-consuming and labor-intensive.
3. **Error-prone:** The manual generation of HTML code increases the likelihood of errors, including typos, formatting inconsistencies, and missing information.
4. **Lack of Consistency:** Ensuring consistency across multiple course pages becomes challenging due to variations in formatting and content presentation.
5. **Need for Automation:** There is a need for a user-friendly tool that automates the process of generating HTML code for course pages, reducing the time and effort required by faculty and administrators.

1.2.Scope Of Project:

This website aims to cover various aspects of course information, including the course title, faculty in-charge, start date, end date, introduction to the course, course objectives, course outcomes, and lesson plan. By incorporating these details, the Course Page Code Generator website streamlines the process of generating HTML code, ensuring consistency and accuracy in course page creation.

2. OVERALL DESCRIPTION:

2.1.Purpose:

The Course Page Code Generator website streamlines the process of generating HTML code for course pages.

2.2.Functionality:

- Intuitive interface for inputting course details.
- Input validation for completeness and correctness.
- Generates structured HTML code for course pages.
- Allows users to preview , save and copy the generated HTML code.

2.3.Operating Environment:

- **Standalone Web-Based Application:** Accessible through modern web browsers.
- **Frontend:** Developed using Angular.
- **Backend:** Built using Node.js with Express.

3. SPECIFIC REQUIREMENTS:

3.1. External Interfaces:

- **User Interface:** The website will provide a user-friendly interface for administrators and faculty members to input course details and interact with the generated HTML code.
- **Backend API:** The frontend application will communicate with a backend API to perform input validation and code generation.

3.2. Functional Requirements:

1.Input Collection:

- Website shall provide separate input forms for administrators and faculty members.
- Form for Administrators contains:
    - Course Title
    - Faculty In-charge
    - Start date
    - End date
- Form for faculty members contains:
    - Introduction to the course
    - Course Objectives
    - Course Outcomes(COs)
    - Lesson Plan → Table form

      1.Unit No.

      2.Topic

3.Lecture Material

4.Lecture Video → Approved Vetting process link.

5.Discourse Link

2. Input Validation:

- Validates input completeness, correctness, and date format accuracy.
- Checks validity of URLs and enforces content length limits.
- Error messages guide administrators and faculty members for accurate data submission correction.
- Lecture video should be approved in the vetting process.

3. Code Generation:

- Upon receiving valid input, the website shall generate HTML code based on the provided course details.
- The generated HTML code will be structured to create a visually appealing and informative course page.
- The lesson plan, presented in table format, shall be accurately mapped to the generated HTML code, ensuring clarity and organization.

4. Output Options:

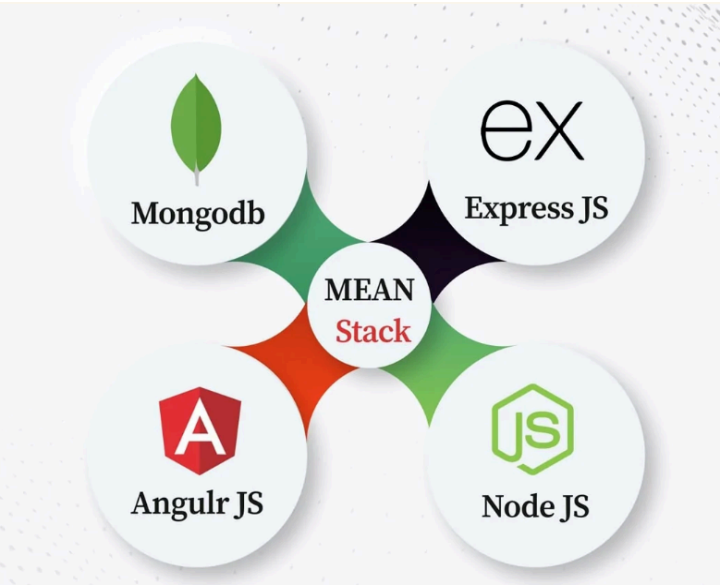
- Administrators will have options to interact with the generated HTML code, including:
    - **Preview:** Administrators can preview the generated HTML code within the application to ensure it meets their requirements before finalizing.
    - **Save:** Administrators can save the generated HTML code as a file for future use or reference.
    - **Copy:** Administrators can copy the generated HTML code to the clipboard for immediate insertion into web pages or documents.

3.3. Non-Functional Requirements:

- **Performance:** The system should generate HTML code quickly, even with a large volume of course details.
- **Usability:** The user interface should be intuitive and easy to navigate, requiring minimal training.
- **Reliability:** The website should be robust and stable, with minimal downtime or errors.
- **Security:** User data should be encrypted and protected from unauthorized access or breaches.
- **Scalability:** It should be able to handle an increasing number of users and courses without significant degradation in performance.

4.TECHNOLOGY STACK → MEAN Stack :

★ **Database System - MongoDB**
★ **Back-end Web Framework - Express.js**
★ **Front-end Framework - Angular.js**
★ **Back-end Runtime Environment - Node.js**



4.1. Frontend → Angular:

- **Angular Components:** Used to create input forms for administrators and faculty members to input course details.
- **Angular Data Binding:** Ensures changes in the UI are reflected in the underlying data model, providing a responsive user experience.
- **Routing in Angular:** Enables navigation between different sections of the application.

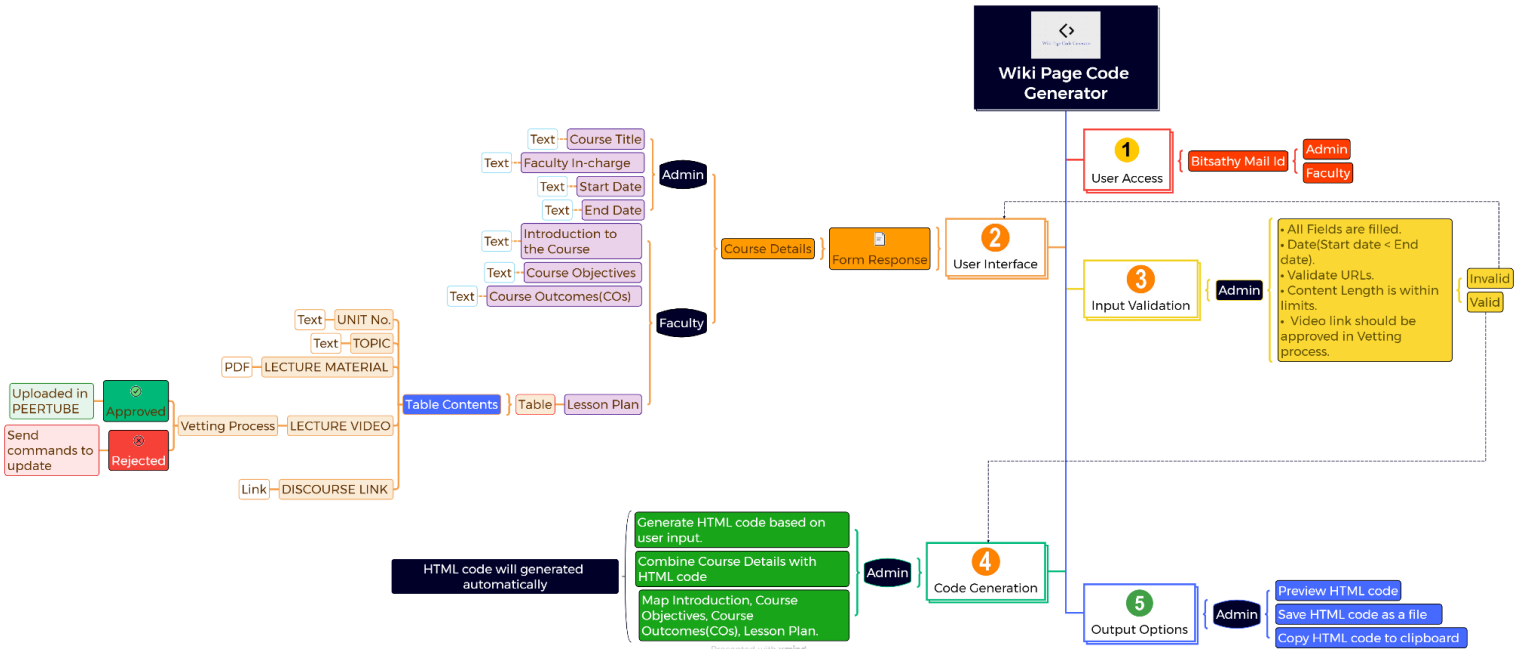
4.2. Backend → Node.js with Express.js:

- **Node.js:** Provides the runtime environment for the backend.
- **Express.js:** Web application framework handling server-side logic.
- **Request Processing:** Receives HTTP requests from the frontend, processes input data, and generates HTML code for course pages.
- **Routing in Express.js:** Manages routing to different endpoints for receiving user input data and serving generated HTML code.
- **Middleware:** Used for input validation to ensure user input is complete and correct.

4.3. Database → MongoDB:

- **Data Storage:** Stores user data, course details, and other application-related information in JSON-like documents.
- **Interacting with MongoDB:** Application interacts with MongoDB to store and retrieve user input data, ensuring persistence and scalability.
- **Flexibility:** MongoDB's flexibility allows for easy storage and retrieval of unstructured data, such as lesson plans stored in a table format.

FLOWCHART:


Wiki Page Code Generator
1
User Access
Bitsathy Mail Id
Admin
Faculty
2
User Interface
Form Response
Course Details
Admin
Text
Course Title
Text
Faculty In-charge
Text
Start Date
Text
End Date
Faculty
Text
Introduction to the Course
Text
Course Objectives
Text
Course Outcomes(COs)
Table Contents
Table
Lesson Plan
Text
UNIT No.
Text
TOPIC
PDF
LECTURE MATERIAL
Vetting Process
LECTURE VIDEO
Uploaded in PEERTUBE
Approved
Send commands to update
Rejected
Link
DISCOURSE LINK
3
Input Validation
Admin
• All Fields are filled.
• Date(Start date < End date).
• Validate URLs.
• Content Length is within limits.
• Video link should be approved in Vetting process.
Invalid
Valid
4
Code Generation
Admin
Generate HTML code based on user input.
Combine Course Details with HTML code
Map Introduction, Course Objectives, Course Outcomes(COs), Lesson Plan.
HTML code will generated automatically
5
Output Options
Admin
Preview HTML code
Save HTML code as a file
Copy HTML code to clipboard

DESIGN:

1.LOGIN PAGE:

2.HOME PAGE: