Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	07 October 2022
Project Name	Unleashing the potential of youth
Maximum Marks	4 Marks

Technical Architecture:

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	E-Learning Module	The core of your app, where users access educational content, courses, and lessons	Front-end: HTML, CSS, JavaScript, React, Angular, or Vue.js Back-end: Node.js, Ruby on Rails, jango, or .NET
2.	Chatbot Integration	It assists users with queries related to courses, content, and general guidance.	Dialogflow, Microsoft Bot Framework, Rasa, or custom chatbot development
3.	Alarm System	Allows users to set alarms or reminders for important events, deadlines, or upcoming classes.	Push Notifications: Firebase Cloud Messaging, Apple Push Notification Service (APNs), or Google Cloud Messaging Scheduler Libraries: Node Schedule, cron jobs, or cloud-based scheduling services
4.	Schedule Designer	Helps users plan study sessions, classes, and assignments, and integrates with the alarm system	- Calendar Components: FullCalendar, Google Calendar API, or custom calendar development - User Interface Tools: React, Angular, or Vue.js
5.	Database	Stores user profiles, course information, chatbot logs, schedules, alarms, and other application data. And manages data retrieval and storage efficiently.	MySQL, NoSQL-(MongoDB, Cassandra, or Amazon DynamoDB)
6.	Content Management System	Provides an admin panel for content creators to manage and organize course materials.	Content Management Frameworks: WordPress, Drupal, or custom CMS

		Allows content upload, editing, versioning, and categorization.	development Admin Panels: React-admin, Flask-Admin, or custom admin interfaces
7.	User Management	Handles user registration, authentication, and authorization. Stores user profiles, preferences, and tracks user progress.	User Authentication: OAuth, OpenID, or custom authentication mechanisms
8.	API layer	Develops APIs to facilitate communication between the front-end and back-end components.	Express.js, Flask, or ASP.NET Web API
9.	Front-end	The user interface of your app, which users interact with.	Web Technologies: HTML, CSS, JavaScript, and popular frameworks like React, Angular, or Vue.js Mobile: React Native, Flutter, or native mobile app development
10.	Analytics and reporting	Integrates tools for tracking user engagement, learning progress, and chatbot performance. Generates reports for administrators and users for data-driven insights.	Analytics Tools: Google Analytics, Mixpanel, or custom analytics solutions Reporting Libraries: D3.js, Chart.js, or custom reporting components.
11.	Testing and Quality Assurance	Involves testing strategies to ensure the app functions correctly, including unit testing, integration testing, and user acceptance testing. Ensures the app is reliable and bug-free.	Testing Frameworks: Jasmine, Jest, Selenium, or custom test suites Continuous Integration (CI) and Continuous Deployment (CD): Jenkins, Travis CI, or GitHub Actions
12.	Monitoring and Maintenance	Implements tools for monitoring the app's performance and resolving issues promptly. Includes regular maintenance and updates to keep the app secure and up-to-date.	Monitoring Tools: Prometheus, Grafana, New Relic, or custom monitoring solutions Issue Tracking: Jira, Trello, or custom issue tracking systems
13.	Cloud hosting	Deploys the app on cloud platforms like AWS, Azure, or Google Cloud to ensure scalability and reliability. Hosts the application on remote servers.	AWS, Microsoft Azure, Google Cloud, or other cloud providers

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Closed Source framework	A closed source framework refers to a software framework whose source code is not freely available to the public. The code is proprietary and typically owned by a specific organization.	Proprietary programming languages, libraries, and tools,c#
2.	Security Implementations	Security implementations refer to the measures and features integrated into a system or software framework to protect it from unauthorized access, data breaches, and other security threats.	Encryption: OpenSSL, Bouncy OAuth, OpenID Connect, JSON Web Tokens (JWT), LDAP (Lightweight Directory Access Protocol). Jenkins
3.	Scalable Architecture	Scalable architecture refers to the ability of a system or framework to handle increased workload or demand by adding resources or adapting to changing conditions without sacrificing performance or reliability.	AWS, Azure, Google Cloud, Nginx, HAProxy
4.	Availability	Availability refers to the extent to which a system or service is operational and accessible when needed.	Apache Kafka, ZooKeeper, Apache Cassandra
5.	Performance	Performance in the context of a framework refers to its speed, responsiveness, and efficiency in executing tasks and handling workloads.	YourKit, performance analysis tools like Intel VTune.Docker,Kubernetes.