



INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
MATERIALS MANAGEMENT DIVISION

Ref. No. (PR.1000042382)

Rfx. No. 6100001913

Scope of work for “Development of the EO portal and Website”.

Sr. No.	Technical Specification	Compliance (Yes/No)	Additional Information if Any
1	1. Project Kick Off (4 Weeks) (Requirements freezing, Design kick-off)		
2	2. Faculty Module (10 Weeks) (SSO, Course Management ,Course Details Management , Course Approval, Approval of Applicants, Feedback) ❖ Faculty Single Sign-On (SSO) Login: * Secure Access: Faculty members can securely log in to the Course Creator Module using their credentials, ensuring data confidentiality and user authentication. *Convenient Authentication: Single sign-on functionality streamlines the login process, allowing faculty to access course management tools seamlessly without the need for multiple credentials. *Course Management: Create New Course: Faculty can effortlessly initiate the creation of new courses, ensuring that all relevant details are accurately captured to provide students with comprehensive course information. * Edit Existing Course: Faculty possess the capability to meticulously curate existing courses,		

	<p>making necessary adjustments to course details and content to maintain alignment with evolving educational standards and objectives.</p> <ul style="list-style-type: none">* Delete Course: Faculty can selectively remove courses that no longer serve educational objectives, ensuring that course offerings remain relevant and up-to-date.* Duplicate Course: Faculty have the option to efficiently replicate existing courses with minor modifications, facilitating the creation of similar courses and streamlining course development processes. <p style="text-align: center;">❖ Course Details Management:</p> <ul style="list-style-type: none">* Set Course Title: Faculty can define the title of the course, ensuring clarity and consistency in course identification.* Set Course Description: Faculty provide a concise overview of the course content, objectives, and learning outcomes, guiding students in their course selection process.* Upload Syllabus: Faculty can seamlessly share detailed course outlines and objectives, providing students with a comprehensive understanding of course expectations and requirements.* Set Learning Objectives: Faculty can clearly outline the goals and objectives of the course, providing students with a roadmap for their learning journey.* Set Prerequisites: Faculty specify any prerequisite knowledge or skills required for enrollment in the course, ensuring that students possess the necessary foundation for success. <p>Set Schedule: Faculty define the course schedule,</p>		
--	--	--	--

	<p>including session timings, duration, and frequency, facilitating effective time management for both faculty and students.</p> <p>* Set Fee Structure: Faculty determine the fees for the course, including payment plans and discounts, ensuring transparency and accessibility in course offerings.</p> <p>* Set Maximum Enrollment: Faculty can limit the number of students allowed to enroll in the course, ensuring optimal learning experiences and resource allocation.</p> <p style="text-align: center;">❖ Course Approval Process:</p> <p>* Submit Course for Approval: Faculty initiate the approval process for new courses, ensuring alignment with institutional standards and objectives.</p> <p>* Track Approval Status: Faculty monitor the status of course approvals in real-time, facilitating transparency and accountability in the approval workflow.</p> <p>* View Feedback from Dean/HOD: Faculty receive and review feedback from relevant stakeholders such as deans or heads of departments, in a chat window sort of a UI</p> <p style="text-align: center;">❖ Approval of Applicants:</p> <p>* Review and Accept/Decline Applications: Faculty have the capability to review and approve or decline applications from students seeking enrollment in the course.</p> <p>* Objective Approval Rules: For objective approval processes, faculty can set specific rules and criteria during course creation. The system</p>		
--	---	--	--

	<p>automatically approves or declines applications based on these predefined rules, ensuring consistency and efficiency in the approval process.</p> <p>* Subjective Review Process: Alternatively, faculty can opt for a subjective review process, where they personally review each application and make individualized decisions based on their judgment and assessment of student qualifications and suitability for the course.</p> <p style="text-align: center;">❖ Creation of Feedback Metrics:</p> <p>Define Feedback Parameters: Faculty can define feedback metrics and parameters during the course creation process, enabling structured evaluation and assessment of student performance and course effectiveness.</p> <p style="text-align: center;">❖ Viewing Feedbacks:</p> <p>* Access Feedback Reports: Faculty can access comprehensive feedback reports, providing insights into student performance, course satisfaction levels, and areas for improvement.</p> <p>* Real-Time Feedback Monitoring: Faculty can monitor feedback data in real-time, enabling timely intervention and course adjustments to address identified issues and enhance the overall learning experience.</p>		
3	<p>3. Learner Module (16 Weeks) (Course Explorer page, Course details page, Application Flow for Specific course, Payment option ,Active courses ,Certificate Management, course feedback)</p>		

❖ **Sign-In:**

* User Authentication: Students can securely sign in to their accounts using their email ID and password, ensuring data confidentiality and user authentication.

❖ **Sign-Up:**

* User Registration: New users can sign up for an account to access course offerings and learning materials.

* Government ID Verification: During the sign-up process, students are required to provide a government-issued identification such as Aadhar or PAN card for identity verification purposes, ensuring the integrity and authenticity of user accounts. The system shall store a hash code of the ID.

* Secure Data Handling: Personal information provided during the sign-up process is handled securely and in compliance with data protection regulations.

❖ **Course Explorer and active course page**

* Display All Courses: A comprehensive list of all available courses should be presented in a user-friendly layout.

* Search Functionality: Users can search for courses using keywords related to the course title, description, or faculty.

* Sort Options: Users can sort courses based on different criteria such as date added, popularity, relevance, and duration.

* Filter Options: Advanced filtering capabilities allow users to narrow down courses by

	<p>categories, subject matter, duration, faculty, price range, and other relevant attributes.</p> <ul style="list-style-type: none">* Course Previews: Brief previews including course name, a short description, and key information (such as start date and fee) should be visible at a glance. <p style="text-align: center;">❖ Course Detail Page</p> <ul style="list-style-type: none">* Course Description: A detailed description of the course content, objectives, and learning outcomes.* Syllabus: A breakdown of the course syllabus, including topics covered in each module.* Faculty Information: Profiles of the faculty members teaching the course, including their credentials and previous experience.* Schedule: Information about the course schedule, including start and end dates, and session timings.* Fee Structure: Clear details on the fee structure, including any available payment plans or discounts.* Prerequisites: Information about any prerequisites needed to enroll in the course.* Multimedia Content: Inclusion of videos, images, and downloadable documents (such as PDFs) to provide additional context and engagement. <p style="text-align: center;">❖ Application Flow and History</p> <ul style="list-style-type: none">* Form Fields: Collecting essential information such as personal details (name, contact information), educational background, and any other required information specific to the course.		
--	---	--	--

	<ul style="list-style-type: none">* Step-by-Step Guidance: A guided process that breaks down the application into manageable steps, helping users to complete it without confusion.* Validation: Real-time validation to ensure all required fields are filled out correctly and to alert users of any errors immediately.* Submission Confirmation: A confirmation message upon successful submission, including next steps and any required follow-up actions.* Application History: A comprehensive list of all previous course applications, providing easy access to past records.* Status Display: Clear indication of the current status of each application, such as pending, approved, or rejected.* Application Details: Access to detailed information about each application, including submission date, course details, and any notes or comments from the reviewing faculty.* Notifications: Alerts or notifications for status changes or additional actions required by the student. <p style="text-align: center;">❖ Payment Option</p> <ul style="list-style-type: none">* Payment Trigger: The option to pay course fees becomes available only after the faculty has approved the application.* Secure Payment Gateway: Integration with secure and reliable payment gateways to handle transactions.* Payment Methods: Support for multiple payment methods, including credit/debit cards, net		
--	--	--	--

<p>banking, and digital wallets.</p> <ul style="list-style-type: none">* Transaction History: A record of all transactions, including receipts and payment confirmations, for future reference.* Automated Invoice Generation: Generation of invoices upon successful payment, which can be downloaded and printed. <p style="text-align: center;">❖ Active Courses Tabular View</p> <ul style="list-style-type: none">* Course Listing: A tabular format listing all active courses that the student is currently enrolled in.* Course Details: Columns providing essential details such as course name, faculty, schedule, and status.* Direct Links: Easy access links to redirect students to relevant resources, such as course materials on Moodle or other integrated learning management systems.* Progress Tracking: Indicators showing the student's progress within each course, including completed modules and upcoming sessions. <p style="text-align: center;">❖ Certificates Management System</p> <ul style="list-style-type: none">* View Certificates: A screen where students can see all their earned certificates.* Download Certificates: Options to download certificates in PDF or other suitable formats.* Share Certificates: Easy sharing functionality to share certificates via email or social media.* Verification Links: Unique verification links for each certificate to allow third parties to validate the authenticity of the certification. <p style="text-align: center;">❖ Course Feedback Submission</p>		
--	--	--

	<ul style="list-style-type: none"> * Feedback Form: A structured form for students to provide feedback on completed courses. Rating System: Options to rate different aspects of the course, such as content quality, delivery, and overall experience. * Comments Section: A space for students to leave detailed comments and suggestions. * Anonymity Option: An option for students to submit feedback anonymously if they prefer. * Follow-Up: Automated responses or follow-up actions based on the feedback provided, ensuring that student inputs are acknowledged and addressed. 		
4	<p>4.HOD & Dean Module (20 Weeks) (SSO , Course Approval)</p> <p>4. 1. HOD Module</p> <p>These features should empower the Head of Department to efficiently manage course approval processes, collaborate effectively with faculty members, and oversee the progress of proposed courses within the educational system.</p> <p style="text-align: center;">❖ HOD Single Sign-On (SSO) Login:</p> <ul style="list-style-type: none"> * Secure Access: The HOD can securely log in to the module using their credentials, ensuring data confidentiality and user authentication. * Convenient Authentication: Single sign-on functionality streamlines the login process, allowing the HOD to access course approval tools seamlessly without the need for multiple credentials. <p style="text-align: center;">❖ Course Approval by HOD:</p>		

Review and Approve/Reject Courses: The HOD has the authority to review courses submitted by faculty members and either approve or reject them based on departmental guidelines and requirements.

* Add Comments to Course Proposals: The HOD can provide comments and feedback on course proposals, facilitating communication with faculty members and ensuring clarity in the approval process.

❖ **Course Timeline Activity:**

* View Timeline of Activities: The HOD can access a timeline of activities for each proposed course, allowing them to track the progress of course proposals and view status changes over time.

* Track Progress and Status Changes: The timeline feature enables the HOD to monitor the course approval process, including submission dates, review periods, and approval decisions, ensuring transparency and accountability in course management.

Apart from the above mentioned features, the HOD can also have the features to create and manage a course, which will require approval from the Dean.

4.2. Dean Module

In the Dean Module, the Dean will have the authority to approve or reject courses submitted by Heads of Departments (HODs). Additionally, they can provide comments on course proposals,

view conversation history, and access the timeline of activities for each proposed course. This comprehensive feature set enables the Dean to effectively oversee the course approval process and ensure alignment with institutional objectives and standards.

❖ **Dean Single Sign-On (SSO) Login:**

* **Secure Access:** The Dean can securely log in to the module using their credentials, ensuring data confidentiality and user authentication.
Convenient Authentication: Single sign-on functionality streamlines the login process, allowing the Dean to access course approval tools seamlessly without the need for multiple credentials.

❖ **Course Approval by Dean:**

* **Review and Approve/Reject Courses:** The Dean has the authority to review courses submitted by Heads of Departments (HODs) and either approve or reject them based on institutional guidelines and objectives.

* **Add Comments to Course Proposals :** The Dean can provide comments and feedback on course proposals, facilitating communication with HODs and ensuring clarity in the approval process.

* **View Conversation History :** The Dean can access the conversation history related to each course proposal, providing insights into previous discussions, decisions, and feedback exchanged during the approval process.

❖ **Course Timeline Activity:**

* **View Timeline of Activities:** The Dean can access a timeline of activities for each proposed

	<p>course, allowing them to track the progress of course proposals and view status changes over time.</p> <p>* Track Progress and Status Changes: The timeline feature enables the Dean to monitor the course approval process, including submission dates, review periods, and approval decisions, ensuring transparency and accountability in course management.</p> <p>These features empower the Dean to efficiently oversee the course approval process, collaborate effectively with Heads of Departments, and make informed decisions to ensure the alignment of proposed courses with institutional objectives and standards.</p>		
5	<p>5. Corporate SPOC Module (24 Weeks) (Course Enrollment , Payment, Participant information)</p> <p>❖ Corporate SPOC Login:</p> <p>* Secure Access: The Corporate SPOC can securely log in to the module using their credentials, ensuring data confidentiality and user authentication.</p> <p>❖ Course Enrollment :</p> <p>* Browse and Select Courses: The Corporate SPOC can browse through available courses and select the ones suitable for their organization's needs.</p> <p>* Enroll for Courses: They can enroll their organization's employees for selected courses, providing details such as the number of participants and preferred session dates.</p>		

	<p>* Individual and Group Enrollment: The Corporate SPOC can enroll employees individually or in groups, streamlining the enrollment process for large-scale training initiatives.</p> <p>❖ Payment Management:</p> <p>* View Course Fees: The Corporate SPOC can view the fees associated with each course, ensuring transparency and clarity in payment requirements.</p> <p>* Make Payments: They can make payments for enrolled courses using secure payment gateways, ensuring timely and hassle-free transactions.</p> <p>* Track Payment Status: The Corporate SPOC can track the status of payments for enrolled courses, ensuring all financial transactions are successfully processed.</p> <p>❖ Participant Information Management:</p> <p>* Enter Participant Details: The Corporate SPOC can enter the names and relevant details of employees who will be participating in enrolled courses.</p> <p>* Edit Participant Information: They can update participant information as needed, ensuring accuracy and completeness of attendee details.</p> <p>* Share Participant Lists: The Corporate SPOC can share lists of enrolled participants with the educational institution or training provider, facilitating seamless coordination and communication</p>		
6	Support Service 12 months		

❖ Additional Scope of Work :

1.UX design

1. Set up a consistent design system and design guidelines
2. Enhancing the user interface, thereby improving the user experience
3. Displaying information as per its importance and hierarchy
4. Personalization, wherever possible
5. Responsive filter functionalities

Design Process

Multidisciplinary team delivering Human-Centered Products

01

02

03

Inspiration

Stakeholder Interviews
User Research
Competitive Analysis
UX Audit

Ideation

Information Architecture
Low & High Fidelity Wireframes
User Interface Design
Interaction Design

Implementation

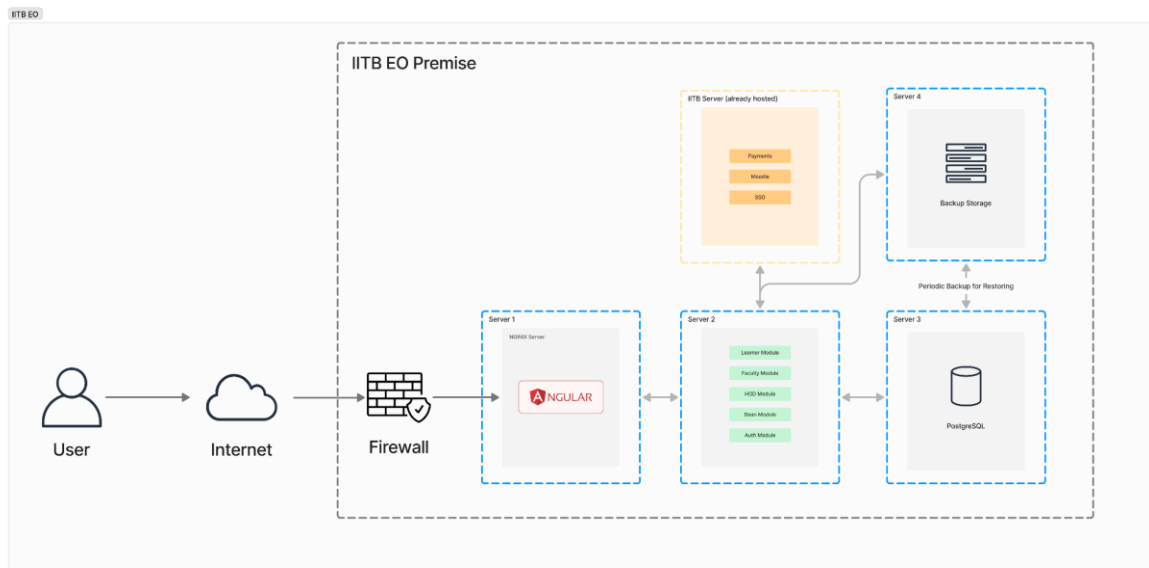
Design Prototyping
Usability Testing and Iterations
Design-Dev Handoff
UI QA

2.Backend design(Design Process)

The backend architecture will handle a large concurrent user base while ensuring a seamless transactional experience. The system will be designed to allow efficient modification, maintenance, and quick release for the respective components. The high-level backend design will consist of the following –

- The components will be detailed using the Domain Driven Design approach, which will allow the creation of the overall architecture which adheres to the business domain. We will build a backend for the frontend layer for effective communication with the presentation layer. It will consume REST APIs to serve as a backend for a frontend.
- The system internally will follow Zero trust architecture to maintain a high level of security between components
- The system will leverage content discovery wherever necessary to provide quick responses to queries on high throughput hot paths.

- Prometheus, and Grafana are to be used for monitoring the components/services and alerting based on the threshold limits
- The entire application will be several unit tests which will be automated during the engagement



Key Components of the Backend Design:

1. User Access:

- **User:** The end user accesses the web application through the internet.
- **Internet:** The user's requests travel through the internet to reach the on-premises infrastructure.
- **Firewall:** This security layer filters incoming and outgoing traffic to protect the internal network from threats.

2. Server Infrastructure:

- **Server 1 (NGINX Server):**
 - **NGINX:** Acts as a web server and reverse proxy that routes incoming HTTP requests to the appropriate services.
 - **Angular:** The front-end application is hosted here, rendering the user interface of the web application.
- **Server 2 (Course Management Service):**

This is the core backend system that manages various modules and functionalities of the application. It includes:

- **Learner Module:** Manages learner-related functionalities.

- Faculty Module: Manages faculty-related functionalities.
- HOD Module: Manages Head of Department-related functionalities.
- Dean Module: Manages Dean-related functionalities.
- Authentication Module: Handles user authentication and authorization for external users.
- **Server 3 (Database Server):**
 - **PostgreSQL:** A relational database management system (RDBMS) that stores the web application data.
- **Server 4 (Backup Storage):**
 - **Backup Storage:** Dedicated to storing backups of the database and possibly other critical data. Periodic backups are taken to ensure data can be restored in case of any failures.

3. IITB Server (Already Hosted):

This is another isolated section that hosts specific services related to the IIT Bombay Educational Outreach. It includes:

- Payment: Manages payment-related functionalities using IITB EO payment gateway..
- Moodle: An open-source learning platform used for delivering educational courses which is hosted in IITB EO premises
- SSO (Single Sign-On): Provides a single sign-on service to allow users to access multiple applications with one set of login credentials which is managed by IITB EO for internal users.

4. Inter-Server Communication:

- **Server 1 and Server 2 Communication:** The NGINX server (Server 1) routes requests to various modules hosted on Server 2.
- **Server 2 and Server 3 Communication:** The application modules on Server 2 interact with the PostgreSQL database on Server 3 to perform CRUD (Create, Read, Update, Delete) operations.
- **Server 3 and Server 4 Communication:** The PostgreSQL database (Server 3) is periodically backed up to the backup storage (Server 4) for recovery purposes. **IITB**

Server Communication: The application also interacts with the IITB Server for services like payments, Moodle integration, and authentication via SSO.

5. Networking and Security:

- The infrastructure is protected by a firewall that ensures only legitimate traffic reaches the servers.
- The communication between different servers and modules is likely secured to prevent unauthorized access and ensure data integrity.

6. Infrastructure and deployment

Server Configurations					
Environment	Instance type	vCPU	RAM	Storage	Use Case
Prod	Linux	8	32	500 GB	Backend Service (EO)
Prod	Linux	4	16	100 GB	Frontend Service
Prod	Linux	8	16	2 TB	Backup Storage (DR)
UAT	Linux	4	16	250 GB	Backend Service (EO)
UAT	Linux	4	16	100 GB	Frontend Service

RDS Instances					
Environment	Instance type	vCPU	RAM	Storage	Engine
Prod	Linux	8	32	1 TB	PostgreSQL
UAT	Linux	4	16	250 GB	PostgreSQL

Assumptions:

- Daily Users Volume: 8-10k
- Concurrent Users: 5-6k
- Moodle is already hosted on another on-prem server
- Bi-weekly file backup of the database will be stored to DR server

Repository Hosting

- **Use a Self-Hosted Git Server:** Instead of GitHub, use a self-hosted Git server like

GitLab, Bitbucket Server, or Gitea. These platforms provide similar features to GitHub but are hosted within your own infrastructure, giving you full control over the source code.

Branching Strategy

- **Keep the Git Flow Model:** The branching strategy based on Git Flow remains applicable. However, ensure that the self-hosted Git platform supports features like merge requests and branch protection.
 - **Main Branch:** Reserved for production-ready code. Should reflect the current state of the live application.
 - **Develop Branch:** For ongoing development work. Features, bug fixes, and other changes should be integrated here.
 - **Feature Branches:** Create from the develop branch, with descriptive names like "feature/feature-name" or "task/task-name."
 - **Hotfix Branches:** For critical production issues, create from the main branch with names like "hotfix/issue-name."
 - **Release Branches:** For final testing and stabilization before merging into the main branch, created from the develop branch and named like "release/v1.0."

Logging and Monitoring

- Logging:
 - **Centralized Logging with ELK Stack:** Use the ELK (Elasticsearch, Logstash, Kibana) stack for centralized log management. Logstash can collect logs from various services, Elasticsearch stores them, and Kibana provides a UI for visualization. This setup is entirely on-premise.
 - **Application Logs:** Ensure your application writes logs to a centralized location accessible by Logstash for aggregation.
- Monitoring:
 - **Prometheus & Grafana:** *Deploy Prometheus for monitoring and alerting. It scrapes metrics from your application and infrastructure. Use Grafana for visualization, offering custom dashboards for key metrics.*
 - **Prometheus Alertmanager:** Set up Alertmanager with Prometheus to handle alerts based on predefined thresholds. Alerts can be routed to different teams via email, Slack, or other channels.

Alerting :

- **Prometheus Alertmanager Configuration:** Configure Alertmanager to notify respective teams when alerts are triggered. Alerts can be routed to different communication channels based on severity and the type of issue.

Visualization and Dashboards:

- **Grafana Dashboards:** Create custom dashboards in Grafana for visualizing key metrics and logs. Grafana supports integration with Prometheus and Elasticsearch, allowing you to monitor logs and metrics in one place for better visibility and troubleshooting.

Additional Considerations for On-Premise Deployment:

- **Infrastructure Management:** Ensure robust infrastructure management tools like Ansible, Terraform, or Puppet are in place to automate deployment, scaling, and updates.
- **Backups and Disaster Recovery:** Implement a backup strategy for both code repositories and infrastructure to ensure data integrity and availability in case of failures.
- **Security:** Secure your on-premise environment with proper network segmentation, firewalls, and access controls. Regularly update and patch your self-hosted tools to protect against vulnerabilities.

7. Web console solutioning

The web console front-end solution will be quick and reactive to the back-end events which are sent to the application tier. They will follow the latest front-end development standards for maintaining quick performance, lesser network calls, and constantly communicating with the backend using web sockets/firebase.

- Using Angular for frontend development for web console applications and super admin features. Higher-order components will be developed for container components at the front end.
- As per the design strategy used by the design team, you may choose to go with component design libraries like Material UI, Bootstrap, and AntD or may choose to develop all the components from scratch.
- Typescript to be used as a standard for strict typing.
- Redux to be used together with Angular as a state management library with proper middleware systems.
- Support for web applications will be restricted to desktop/large screens only.
- Browser support:

- Chrome - 100+
- Firefox – 100+
- Safari(Mac) - 14+
- Edge

8. Quality assurance

Service Provider Should strive to assure the Confidentiality, Integrity, and Availability of Service Provider customers' and other stakeholders' information & assets from any internal, external deliberate or accidental threats. In line with IITB EO policy, the information security management system is integrated with the organization's processes and overall management structure and committed to full implementation by —

- Service Provider applies the risk management process by giving confidence to interested parties that risks are adequately managed.
- Service Provider Should ensure client data/resources remain safe and secure during transmission, handling, and processing.
- Service Provider Should commit constant maintenance and evaluation for Continuous Improvement.
- Service Provider Should have more detailed information security policies, procedures, standards & guidelines.
- Service Provider Should be committed to communicating the importance of effective information security management.
- Information security is deemed to safeguard three main objectives
 - Confidentiality — data and information assets must be confined to people authorized to access them and not be disclosed to others.
 - Integrity – Keeping the data intact, complete, and accurate and IT systems operational.
 - Availability – an objective indicating that information or system is at the disposal of authorized users when needed.
- Most of our work is to be done with a zero-trust policy mindset. Access controls are stringent, with the least access given by default. All-access controls are kept under an approval process per Project. As far as possible, try to work in the client environment with limited access to non-PI data. All dev servers are kept under restricted VPNs.
- Service Provider Should complete end-to-end stress testing for the application for a high level of concurrent access. Tools to be used like JMeter or similar to create concurrent access and perform load testing on various features.

- Consistent logging and reporting of bugs will be conducted to release a defect-free product to the end consumer.
- Information required for auditing purposes and compliance will be tracked and stored to make them available when needed by the regulatory authorities.

9.Security

To be implemented security across various levels of the product and the team working on the product. This will ensure to meet the required security and regulatory compliance about the US healthcare standards.

- Human Resource
- All team members involved in the product engineering effort will be onboarded with a complete understanding of the relevant processes.
- All the Service Provider Should team members sign an NDA at the time of joining and are bound to maintain the confidentiality
- Service Provider Should is a SOC2 Type 2 compliant organization
- Workspace Assets and Infrastructure
- All assets provided to the individuals will be encrypted and can be accessed with secure logins
- All machines' access will be monitored and logged Every resource will have secure access to the workspace via a face scan access control mechanism
- Product Security
- The app will allow secure access to the product features with strong password-based authentication
- The app will allow access to the features based on the role-based access control mechanism
- The app will allow access to the features based on the role-based access control mechanism
- All the data stored locally within the app will be encrypted and will be sent to the backend using Secure connections using SSL/TLS protocols
- All server access will be through VPN and Bastion Hosts only, and all the accesses were logged and tracked
- Integrate security tools (e.g., SAST/DAST/SCA) such as implementing Checkmarx/Veracode security scans into the CI/CD pipeline

- All open-source libraries used for product development will be scanned for vulnerabilities.

10. Delivery and Maintenance of Solution

Various team members with various skills will be onboarded during the product's lifecycle. They will be providing for the below-mentioned capabilities on a high level

Team should follow the Agile development methodology for tracking the execution of the project. As a part of the overall software development process, team should follow the below-mentioned best practices for software delivery -

1. We would be onboarding the right balance of developers, quality assurance engineers, product managers, and project managers to ensure we have the required skills to deliver the desired outcomes. Clean coding and clean architecture design based on the experience of building large-scale web apps
2. Following OWASP-based security protocols while implementing the product features and using tools like CheckMarx for application security testing
3. Automating the testing processes to achieve 90% test coverage using various tools. All bugs are tracked and closed before releasing the build to the UAT environments. This will ensure we can capture all issues during the internal QA process.
4. Automated deployments using CI/CD pipelines using tools like Jenkins.
5. We will adopt the Agile Development methodology, where we will iteratively develop the product artifacts with separate sprints running in parallel for Design, and Product Development
6. The design sprint will start during the discovery phase and will be 1 sprint ahead of product development sprints.
7. The project execution will be divided into two-week sprints and the generation of various sprint reports to track progress and productivity. We recommend using tools like JIRA for tracking tasks and resource productivity. For product management and understanding the domain, we recommend using tools like Confluence for documenting business logic and technical decisions around approaches, architecture, and infrastructure documents. This will ensure in having the latest and up-to-date documentation always being available for current and future team members.
8. With a Scrum approach, we adopt showcasing constant progress, ensuring all

timelines are met as per the initial commitment with the client. For communication team will be using a variety of tools like Slack, MS Teams, Emails, etc.

9. Quality is an integral part of our commitment to our clients. To ensure the same, we follow processes like Business-Driven Development in automating our testing, enabling us to verify and detail all our requirements.

B. Kindly ignore Point No. 6 of section 3 of NIT Document. The below Payment Terms and Conditions will be applicable).

- **Payments Terms and Condition :**

The payments will be based on achievement of Milestone Basis

Sr No	Milestone	Timelines	%
1	Project Kick-off	4 Weeks	10%
2	Faculty Module	10 Weeks	35%
3	Learner Module	16 weeks	25%
4	HOD & Dean Module	20 Weeks	20%
5	Corporate SPOC Module	24 Weeks	10%

The Payment will be release as per Milestones and only after certification from the Indentor through NEFT/RTGS.

The Service provider must agree to maintain confidentiality regarding all property information and content, moreover all intellectual property rights will be retained by IIT Bombay.