**Requirement Gathering and Analysis Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 06-07-2024 |
| Team ID | PNT2022TMID SWTID1720175019 |
| Project Name | Nexus Learn – Online learning platform |
| Maximum Marks |  |

**Solution Architecture:**

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

* Find the best tech solution to solve existing business problems.

The goal of this solution architecture is to bridge the gap between business problems and technology solutions for an online learning platform focused on various technologies. This platform will allow users to both teach and learn courses. The objectives are to identify the best tech solutions to solve existing business problems, describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders, define features, development phases, and solution requirements, and provide specifications for defining, managing, and delivering the solution.

To address user engagement, the platform's frontend will be built using React.js to ensure a dynamic and responsive user interface. Gamification elements such as badges, leaderboards, and progress tracking will be incorporated to motivate users. Additionally, WebRTC will be used to enable real-time interaction for live classes, ensuring a highly engaging user experience.

Content delivery is a critical aspect of the platform. Amazon S3 will be utilized for scalable storage of multimedia content, ensuring that all course materials are stored efficiently.

Scalability is essential for accommodating growth in both users and content. Backend services will be deployed on AWS EC2, which offers auto-scaling capabilities to handle increased demand. MongoDB Atlas, a fully managed cloud database service, will be used to support scalable data storage and retrieval, ensuring the platform can grow without performance degradation.

Security is a top priority for protecting user data and ensuring secure transactions. HTTPS will be implemented for secure data transmission, and JWT will be used for secure user authentication and authorization. Encryption will be employed to protect sensitive data, and regular security audits will be conducted to ensure ongoing security assessments and compliance with industry standards.

* Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

The solution architecture is structured into several key components. The frontend, built using React.js along with HTML/CSS and either Bootstrap or Tailwind CSS for a responsive and consistent design, will handle user registration, course browsing, content viewing, interactive lessons, quizzes, and discussions. The backend, developed with Node.js and Express.js, will manage user information, course content, payment processing, and analytics, providing API endpoints for frontend interactions. MongoDB will serve as the database, storing user profiles, course data, payment records, and analytics information. Amazon S3 will be used for storing videos, documents, quizzes, and other multimedia content. The backend and storage services will be hosted on AWS, with EC2 providing scalable backend deployment, CloudFront ensuring fast content delivery, RDS handling relational database needs, and Elastic Beanstalk managing application deployment.

* Define features, development phases, and solution requirements.

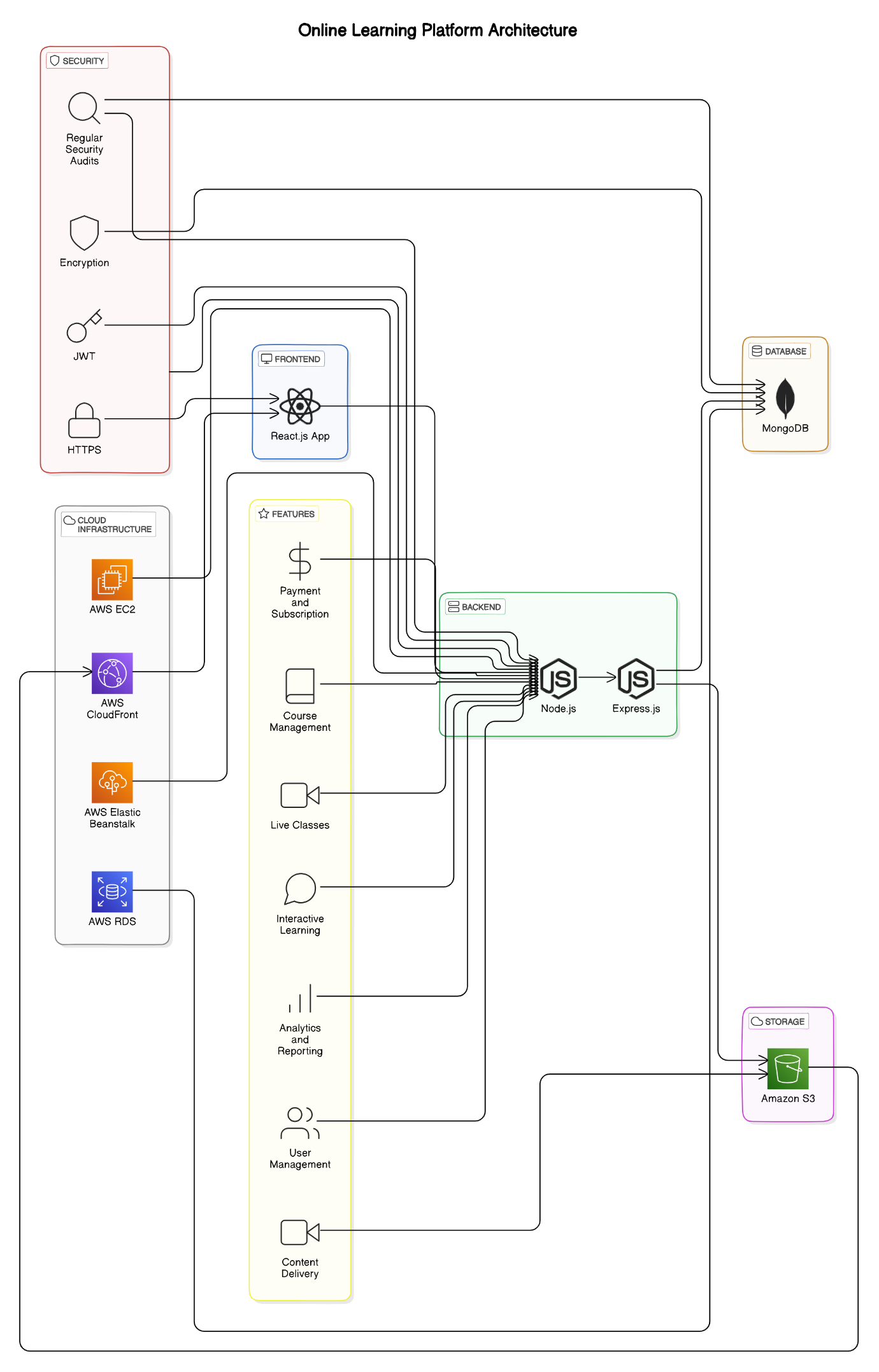
The platform will feature comprehensive capabilities, including user management, course management, content delivery, interactive learning, live classes, payment and subscription management.

Development of the platform will proceed through several phases. Planning and requirement analysis will involve gathering detailed requirements and defining the project scope and objectives. The design phase will include creating detailed solution architecture, designing the UI/UX for the frontend, and designing the database schema. Development will cover frontend development with React.js, backend development with Node.js and Express.js, setting up MongoDB, and integrating AWS services. Testing will encompass unit, integration, user acceptance, security, and performance testing. Deployment will involve deploying the application on AWS using EC2, S3, CloudFront, and Elastic Beanstalk, with auto-scaling and load balancing configured. Maintenance and updates will include regular updates, feature enhancements, continuous monitoring, and security audits.

* Provide specifications according to which the solution is defined, managed, and delivered.

Specifications for each component of the architecture are as follows. The frontend will be built with React.js, Bootstrap or Tailwind CSS for responsive design, and HTML/CSS for structuring and styling. The backend will be developed using Node.js for scalable operations and Express.js for web application logic and API endpoints. MongoDB will be used for flexible and scalable data storage. Amazon S3 will provide reliable and scalable multimedia content storage. AWS EC2 will support scalable backend services, CloudFront will ensure fast content delivery, and Elastic Beanstalk will manage application deployment. Security will be ensured through HTTPS for secure communication, JWT for secure authentication and authorization, encryption for data protection, and regular security audits to identify and mitigate vulnerabilities.

**Example - Solution Architecture Diagram:**

****