**Requirement Analysis Document**

**1. Project Overview**

* **Project Name:**
  + TechSpace Academy
* **Purpose:**
  + The project is about e-learning and the purpose of this e-learning platform is to provide an interactive, engaging, and accessible online learning environment for users, allowing them to acquire knowledge and skills at their own pace.

**2. Objectives**

* Enhance learning experience by offering interactive content.
* Provide a flexible learning schedule to accommodate diverse online learners.
* Track learner progress and provide feedback to improve retention and performance.

**3. Team**

* Team Members: Aditya Mhatre (Team Leader), Geetha Narasimha (Assistant Team Leader), Sandeep parwal, Taral Kotwal.
* BVC accounts to be used - Aditya Mhatre and Sandeep Parwal
* Communication: We have a channel called TechSpace Academy in Teams and use this to communicate anything regarding the project.
* Meetings: As of now we have decided to have Daily standup to discuss on everyday progress, once in two weeks(for a sprint) we will have Grooming session to discuss upcoming tasks and the outcome expected out of the tasks. Planning session to clarify any doubts on the groomed tasks and assign them between team members, Review session to check if each task is completed with the expected output.
* Repository: We are using the existing project from which we will modify and build as per the requirements. We are using Github to keep out project code base and maintain the version, merging everyone’s changes and maintain the code base without losing any code by overwriting.

**4. Scope**

* **Inclusions:**
  + User registration and profile management.
  + Course creation tools for instructors.
  + Multimedia content support (videos, PDFs, quizzes).
  + Mobile-friendly design.
* **Exclusions:**
  + Offline access to content.
  + Integrated payment processing for paid courses.
  + Advanced analytics for institutional users (initial phase).

**5. Target Audience**

* **Primary Users:**
  + Students ranging from high school to adult learners seeking professional development.
  + Educators looking to expand their teaching methods and reach.
  + Corporations seeking training solutions for employees.

**6. Functional Requirements**

* **User Roles:**
  + **Admin:** Manages users, courses, and site settings.
  + **Instructor:** Creates and manages courses, grades assignments.
  + **Learner:** Enrolls in courses, accesses content, submits assignments.
* **Features:**
  + **User Registration/Login:** Secure signup and login process.
  + **Course Creation and Management:** Tools for instructors to build courses.
  + **Course Listing and selection:** Feature to show the available courses.
  + **Course Details:** Feature to detail the particular course.
  + **Content Upload:** Support for various file types (videos, documents, quizzes).
  + **Certification Generation:** Automated certificates upon course completion.
  + **Offline Payment Processing:** Status update Integration with selected courses by students.

**7. Non-Functional Requirements**

* **Theme:**
  + Educational images at the background and with blue colour
* **Navigation:**
  + Keep the navigation links at the Header part
* **Logo placement:**
  + Keep the Logo at the top left corner of the page
* **Content Listing:**
  + Going away from typical listing of data, we are planning to have div boxes with images for each course which take us to course details page onclick

**8. Technical Requirements**

* **Platform:**
  + Web-based application compatible with major browsers (Chrome, Firefox, Safari).
  + Mobile app development for iOS and Android platforms.
* **Technology Stack:**
  + Frontend: HTML5, CSS3, JavaScript (React.js).
  + Backend: PHP to implement CRUD operations and communicate with Database.
  + Database: MongoDB or PostgreSQL or MySQL.
  + Hosting: AWS Cloud services.
* **Hosting/Infrastructure Requirements:** Scalable cloud hosting solutions to accommodate growing user numbers and features.
  + Amazon S3: It is used to store static contents; we will have static pages and images or videos which will be stored here
  + EC2 instance: As of now we will be using EC2 instance to deploy our project and launch it assuming that our project to be midlevel budgeted.
  + VPC: We can run code, store data, host websites, and do anything else they could do in an ordinary private cloud, but the private cloud is hosted remotely by a public cloud provider. It provides isolation on data.
  + RDS: We will use Amazon RDS which is a managed relational database service for our data content of the project. User information(Student, Admin), Course information will be stored in this which is used for various operations.
  + Terraform: We will be using Infrastructure as code to automate the provisioning of certain infrastructure like EC2 servers, S3, RDS, firewall, policies etc
  + AWS Codepipeline: We may be using AWS codepipeline to automate continuous delivery pipelines for fast and reliable updates.
  + CloudWatch: We may use Cloudwatch to monitor the complete stack (applications, infrastructure, network, and services) and use alarms, logs, and events data to take automated actions. This frees up important resources and allows you to focus on building applications and business value
  + CloudFront: It is a content delivery network (CDN) service that helps you distribute your static and dynamic content quickly and reliably with high speed. We may use this to Securely deliver our project content with low latency and high transfer speeds
  + Application Load balancer: We will use the Load balance which automatically distributes the incoming application traffic across multiple instances of the application. We will obtain security, autoscaling, high availability using this.
* **Other Tools:** 
  + JIRA: We would be using JIRA for task management within the team and track the status and completion of the tasks every sprint.
  + Github Actions or Jenkins: We may use to automate certain workflow to merge code changes from feature branch to main branch and to track the quality checks like vulnerabilities/code coverage etc.

**9. User Interface Requirements**

* **Wireframes:**
  + Include wireframes for key pages: Home, Course List, Course Detail, User Dashboard, and Admin Panel.
* **Design Guidelines:**
  + Color palette: Clean, modern colors that enhance readability.
  + Typography: Use of legible fonts for headings and body text.
  + Responsive design to ensure compatibility with all devices.

**10. Testing Requirements**

* **Types of Testing:**
  + **Unit Testing:** To verify individual components.
  + **Integration Testing:** To ensure different modules work together.

**11. Glossary**

* **LMS (Learning Management System):** Software for delivering and managing educational courses.
* **UAT (User Acceptance Testing):** Testing phase where real users validate the system before launch.

**Conclusion**

This requirement analysis document outlines the essential components needed for the successful development of the e-learning platform. By addressing the needs of stakeholders and defining clear objectives, the project aims to create an enriching educational environment that is accessible to all users.

**Sprint Timeline**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cycle** | **Start Date** | **End Date** | **Week1** | **Week2** | **Week3** |
| Sprint 2 | 21-10-2024 | 01-11-2024 | \* Freeze the features and remove the unwanted functionalities from the Admin, Customer, Registration/Login and Home pages \* Design Database, freeze DB and table structure and relationship, Normalize and document \* Finalize the logo, theme, footer information, header information \* Update RAD | \* Create the DB schema, tables using DDL statements and link the relationship as per the design. \* Implement the Logo, theme, header and footer templates  \* Create/Edit/Delete Customer information \* Update RAD |  |
| Sprint 3 | 04-11-2024 | 15-11-2024 | \* Modify Course list page according to the requirement \* Modify course details page according to the requirement \* Modify Contact us page  \* Modify Admin panel/Registration/Login page \* Update RAD | \* Create/Edit/Delete Course metadata(course table info) \* Create/Edit/Delete course details information and link with course table and any other tables. \* Send contact us information and store in the table. \* Complete any front end changes or finetuning is required to make all the webpages consistent and beautiful \* Update RAD |  |
| Sprint 4 | 18-11-2024 | 06-12-2024 | \* Integrate the frontend with backend \* Make sure all the actions from frontend are working exactly and calling necessary backend functionalities. \* Make sure the exact data is getting stored or reflecting according to the operation used \* Write unit test cases to test the overall fucntionalities \* Update RAD | \* Integration test, Regression test and run the application in every individual system locally and test. \* All feature branches are merged and verified \* Create EC2, S3, RDS using terraform \* Create necessary security group, VPC, rules and policies and link accordingly \* Create Codepipeline to automate the git pull into server and reflect the changes in EC2. \* Deploy the project in EC2 and test the application using the cloud URL \* Update RAD |  |

**Meeting Occurrences**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Grooming session** | **Planning session** | **Week1 Review** | **Week2 Review** | **Week3 Review** |
| Sprint 2 | 21-10-2024 | 21-10-2024 | 25-10-2024 | 01-11-2024 |  |
| Sprint 3 | 30-10-2024 | 04-11-2024 | 08-11-2024 | 15-11-2024 |  |
| Sprint 4 | 13-11-2024 | 18-11-2024 | 22-11-2024 | 29-11-2024 | 06-11-2024 |
| Daily Stand up | | | Monday – Friday(Every weekday) | | |

**Github Project link:** [**Click Here**](https://github.com/GeethaNarasimha2510/TechSpace-Academy)

**Team Members: Group 2**

**Aditya Mhatre** - [a.mhatre623@mybvc.ca](mailto:a.mhatre623@mybvc.ca)

**Geetha Narasimha** - [g.narasimha064@mybvc.ca](mailto:g.narasimha064@mybvc.ca)

**Sandeep Parwal** - [s.parwal175@mybvc.ca](mailto:s.parwal175@mybvc.ca)

**Taral Kotwal** - [t.kotwal365@mybvc.ca](mailto:t.kotwal365@mybvc.ca)

**Artifacts**

**AWS Architecture Diagram**

Following are the services which we are thinking to use for our project:

Server: AWS, Linux/Ubuntu,

Programming Language: PHP

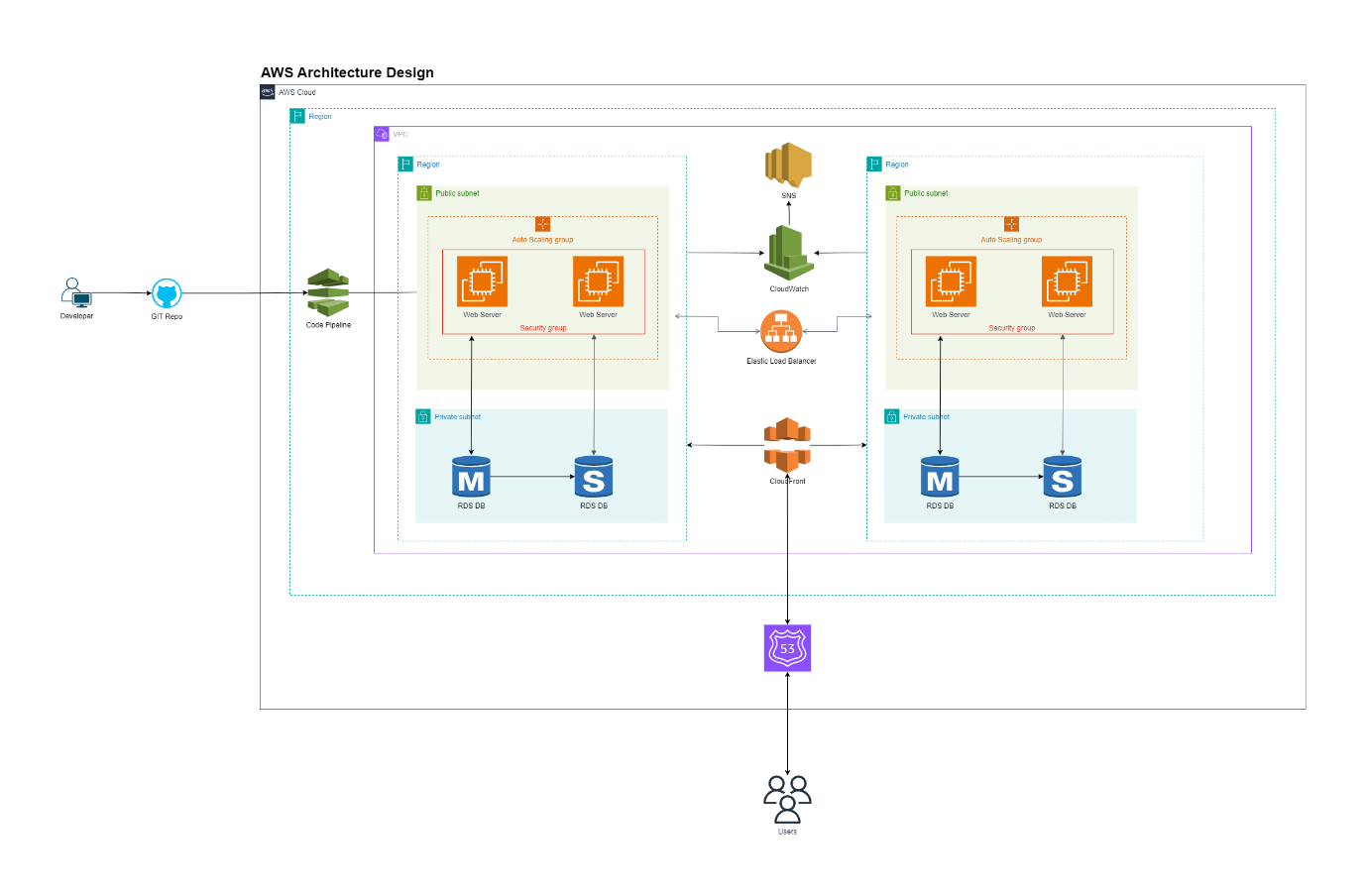
Scripting Language: HTML / CSS

Database - MySql

AWS Services: EC2, S3, Route 53, RDS - MySql, CodePipeline

Other Services: Terraform

Code Repo: GitHub



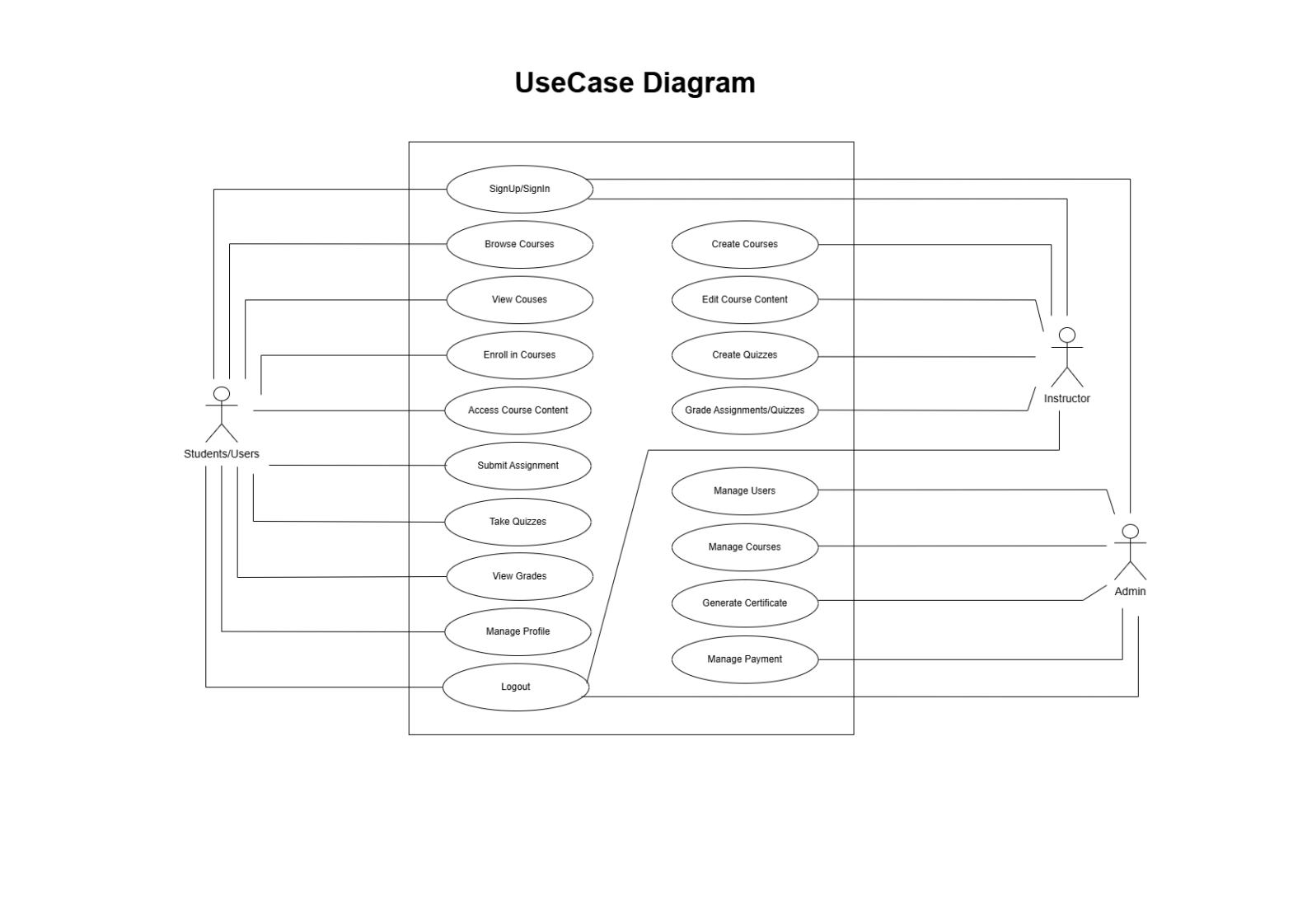
For our Tech Academy online project, we have carefully chosen services and tools that together offer a secure, scalable, and efficient environment, ensuring robust functionality for both administrators and users. Here’s a brief description of our selections:

- AWS Services:   
   - EC2 – For hosting our PHP applications on a scalable, customizable virtual server.  
   - S3 – To securely store and serve static files, such as course materials and multimedia.  
   - Route 53 – For domain name management, ensuring that users can access the website reliably and with optimal performance.  
   - RDS (Relational Database Service) – To manage our MySQL database, offering a scalable, fully-managed solution.  
   - CodePipeline – For implementing a CI/CD pipeline, automating the build, test, and deployment phases, and enabling smoother updates.

- Other Services: Terraform – We’re using Terraform as an Infrastructure-as-Code tool, enabling us to automate and version control our infrastructure setup and provisioning across multiple environments.

- Code Repo: GitHub – GitHub will allow us to manage code with version control, collaborate efficiently, and track changes, providing a reliable system for team development and code management.

This combination ensures a solid, integrated platform that supports high performance, scalability, and ease of maintenance, delivering a seamless experience for our users and course administrators alike.



Student/User connects to:-

* SignUp/SignIn
* Browse/Courses
* Enroll in courses
* Manage Profile
* Logout

Instructor connects to:-

* SignUp/SignIn
* Browse/View Courses
* Upload Course syllabus
* Manage Profile
* Logout

Admin connects to:-

* SignUp/SignIn
* Manage Users
* Manage Courses
* Add Course
* Logout

**ER diagram**

**A diagram of a student's program

Description automatically generated with medium confidence**

**Assumptions:**

* As part of the project, we are making some content dynamic where values are added in frontend, and they are processed and stored in DB via backend. Accordingly, we have designed above tables with Normalization applied and necessary attributes with data types.
* **User entity:** It’s a master table whereUser details when registered who can be Student, Admin, Instructor will be stored in this table with required details.
* **Course entity:** It’s another master data which will have the course and its details like what is the course name, overview, fees, which instructor is mapped, department, course level, ratings etc. This can be added by only Admin as of now.
* **Course curriculum:** This table consists of the curriculum or syllabus of the course added in Course table. Each course can have more than one syllabus and its details like topic and its description. We may add few more contents as required. This data can be added either by Admin or Instructor as well.
* **Studentenrollment:** This table consists of the data of student and course enrolled. Any student who is registered in User table can search for required course and enrol for the course. One student can enrol more than one course, and one course can be enrolled by more than one student too. It’s a n:m relationship. Once enrolled, this information can be viewed by Admin and Instructor too.
* For all the data, we are allowing it to be created and viewed only. Except for studentenrollment which needs to be approved either by Instructor or Admin, but other data will not be updated or deleted. We may introduce update or deletion in future.

**Frontend Changes**

As we have the available template for the project, some of the frontend pages needed to be cleaned up because the content is more than what is expected in the project requirement catalogue. Mainly for sprint 3, Home page and Course list page were targeted to cleanup either by editing the content or removing few sections. Also we are in progress in making the common content to be consistent across all the pages.

A screenshot of a website

Description automatically generated

A screenshot of a web page

Description automatically generated