**Topic: Online Learning**

**+ Introduction**   
This project is concerned with the analysis, design, development, implementation, and evaluation of an online learning management system to provide a user-friendly environment for prospective users to acquire knowledge at any educational level and to bridge the gap between administrators and users.

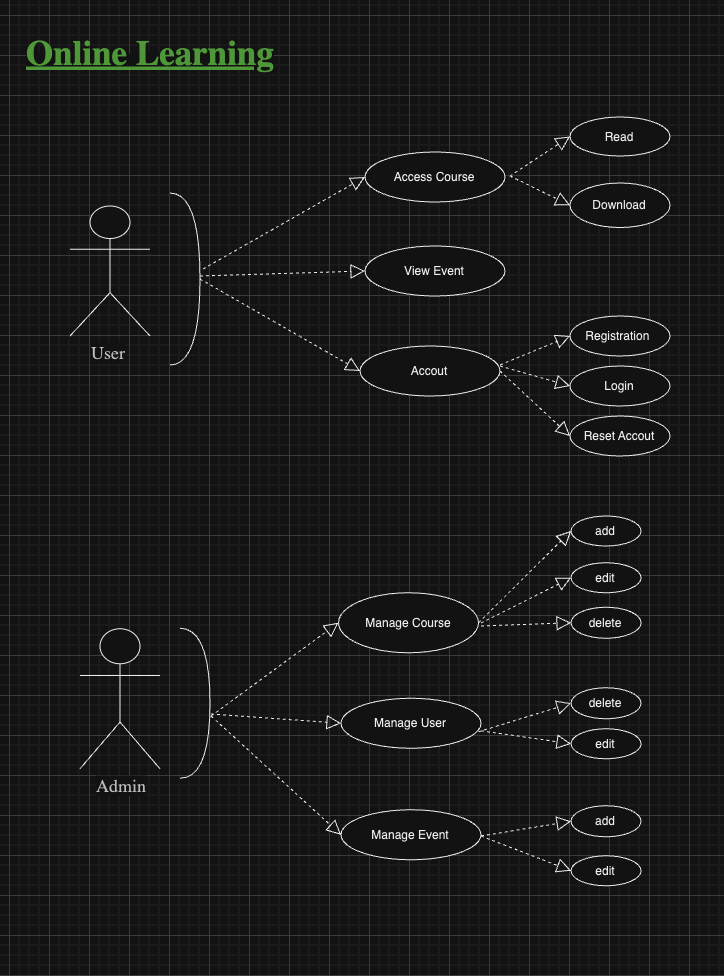
**+ Functional Requirements (Backend Design)**The main actors in the use of the proposed application will be the administrator, teacher, and student. In this online learning system, you will be able to add a student to a class and download a file, a course, a subject, and a video. This project will contain a lot of advanced modules which makes the backend system very powerful. The list of requirements for this project was also influenced by the work done in the background research section above.    
  
PROPOSED MODULES:   
below is a description of the modules proposed for the online learning system.

|  |  |  |
| --- | --- | --- |
| **Module** | **Objective** | **Features** |
| Activity logs | Manage activity logs | * Admin can manage the activity logs. * Admin can edit/delete the activity logs. * Admin can see the list of all activity logs. * Admin can see his activity logs |
| Class | Tracks all the information and details of classes. | * Admin can add new classes. * Admin can see the list of classes details. * Only admin can edit and update the record of the classes. * Admin will be able to delete the records of the classes. * All classes forms are validated on client side using JavaScript |
| Subjects | Administrator can add subjects and a teacher can see subjects | * Admin can manage the subjects. * Admin can edit/delete the subjects. * Admin can see the list of all subjects |
| Users | Tracks all the information and details of the users. | * Admin can see the list of student’s details. * Only admin can edit and update the records of the users. * Admin will be able to delete the records of the users. * All Users publication forms will be validated on the client side using JavaScript. |

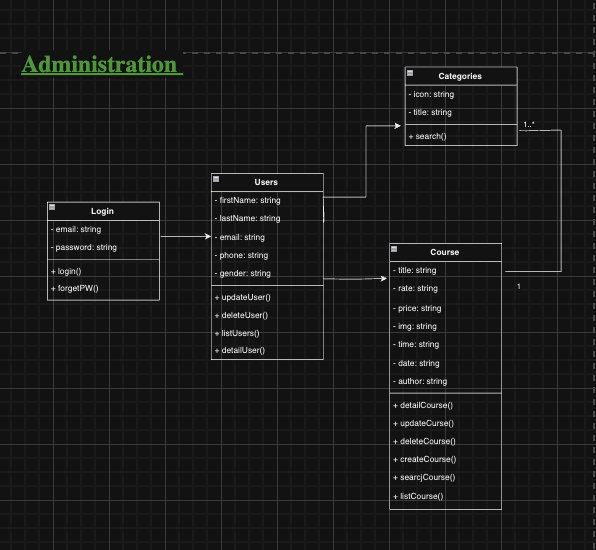
below gives a description of the functional requirements for each user:

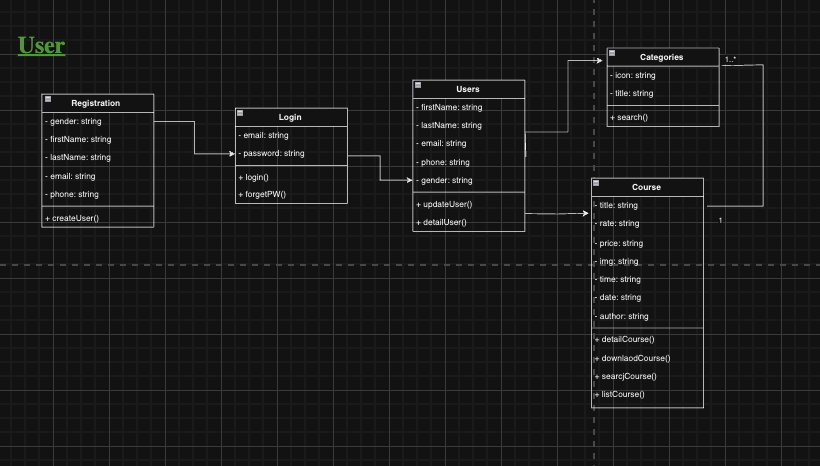
|  |  |
| --- | --- |
| **User** | **Functional Requirements** |
| Admin | * Login for Admin * Forgot password for Admin. * Edit Profile for Admin * Change password for Admin. * Logout functionality * Dashboard for Admin User * **MANAGE Course:** Add new, edit existing, view details of a course and see a listing of all courses. * **MANAGE FILES**: Add new files, edit existing files, view details and a listing of all files. * **MANAGE User**: Edit the existing student, view details of the user and see a listing of all users. * **MANAGE SUBJECT**: Add a new subject, edit the existing subject, view details of a subject and see a listing of all subjects. |
| User | * **User Registration**: Any users can register on the website using the registration module. * **User Login**: This is the login form from where users can login to the system. * **User Change password**: A user can change his account password from this module. * **User View Announcement**: Users can view announcements. * **User View Course**: Users can view document and download. |

Shows the use case diagram:



Shows the Domain Model UML:





Shows the ERD (Entity Relational Diagram):

**+ Microservice Architecture:A diagram of a application

Description automatically generated**

**+ Technologies for Development:**

* Languages: React (ReactJS and ReactNative) and NodeJS (ExpressJS)
* Database: MongoDB
* Database UI: MongoDB compass
* Authentication: JWT
* Testing: Jest
* Deployment: Docker, AWS
* Testing tool: Postman
* GitHub: Coding Management
* CI/CD: Project Pipeline and auto-deployment