Online Learning Platform using MERN

- **Abstract**

This project report presents an Online Learning Platform – CodeVector designed to enable flexible, accessible education for students while providing instructors with a robust set of tools to manage courses. The OLP offers login and course management for instructors, course enrollment and certification for students, and an admin console for overall oversight. Developed with React, Bootstrap, Express, and Node, the platform integrates secure access, video storage via Cloudinary, and email notifications using Nodemailer.

- **Introduction**
    - **Purpose of the Platform**: The purpose of the OLP in promoting accessible, flexible, and high-quality online learning.
    - **Problem Statement**: The platform addresses the need for an integrated learning environment where students and instructors can interact, share knowledge, and track learning progress in a secure, user-friendly application.

- **Key Features of the Online Learning Platform**
    - **User Roles and Access Control**
        - **Instructor**: Instructors can log in, create and manage courses, and view enrolled students.
        - **Student**: Students can sign up, view courses, enroll, track progress, and receive certifications.
        - **Admin**: The Admin has access to student and instructor data, can view course listings, and manage platform content.

- **Instructor Functionalities**
    - **Course Creation**: Instructors can create courses with banner images, prerequisites, and multiple sections.
    - **Content Upload**: Video content is stored on Cloudinary, optimizing media management and streaming.
    - **Student Management**: Instructors can view the list of students enrolled in each course, facilitating student monitoring and engagement.
    - **Course Management**: Instructors can edit and delete courses, ensuring the platform remains updated and relevant.
- **Student Functionalities**
    - **Course Exploration and Enrollment**: Students can browse available courses, view course details, and enroll in courses.
    - **Course Progress and Certification**: Upon course completion, students are presented with a certificate that includes their name, course name, and date of issue.
    - **Discussion Forum**: Each course includes a discussion section where students can initiate or join threads, fostering a collaborative learning environment.
    - **Profile Management**: Students can edit their profile data, enhancing personalization within the platform.
- **Admin Console**
    - **User Management**: Admins can view detailed profiles of students and instructors.
    - **Course Oversight**: Admins can monitor all courses' basic details and delete any course that violates platform policies or becomes obsolete.

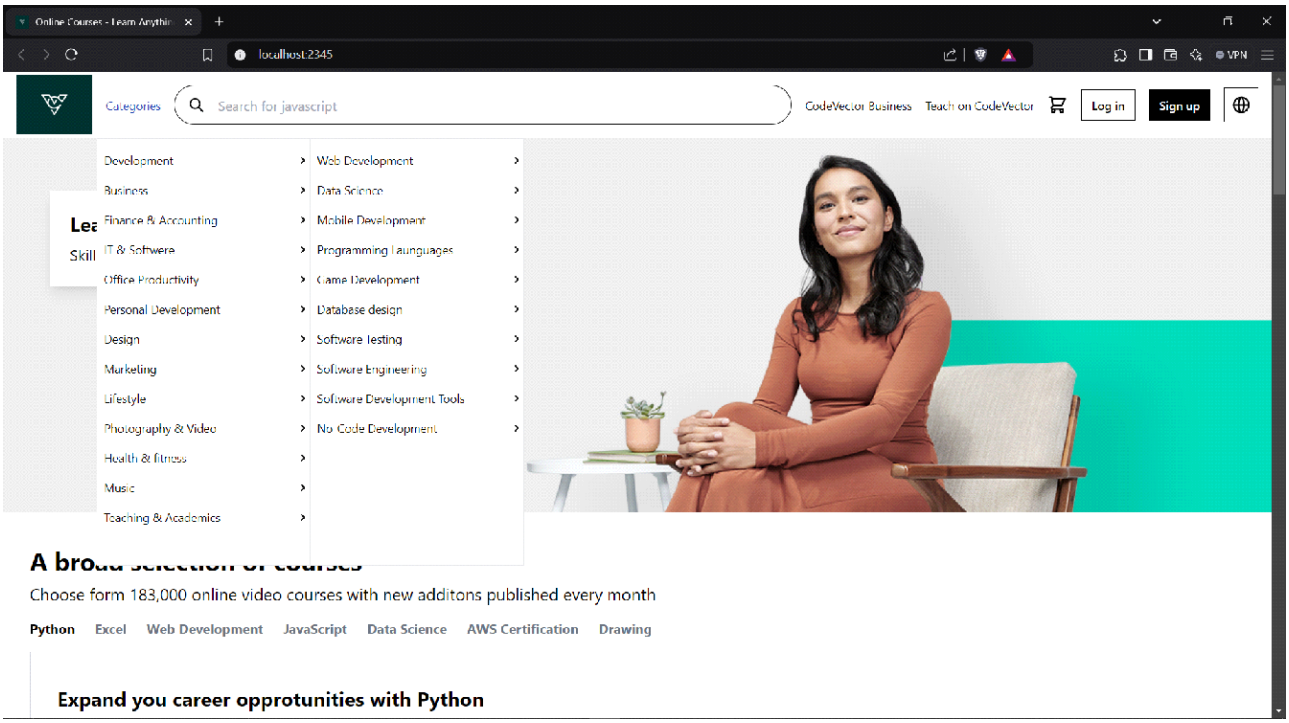
- **Technical Architecture**
  - **Frontend**
    - **Framework**: React is used for building a responsive and interactive user interface.
    - **UI Components**: Bootstrap is used alongside custom CSS to create a user-friendly and visually appealing interface.
    - **Protected Routes**: All sensitive pages are protected using JWT tokens to ensure secure access control.
  - **Backend**
    - **Framework**: Node.js and Express handle server-side logic, API integration, and secure communication between client and server.
    - **Authentication and Security**: JWT tokens are used for authentication, ensuring that only authorized users can access protected routes.
    - **File and Media Storage**: Cloudinary is integrated to store video files, enabling efficient media handling for course sections.
    - **Email Integration**: Nodemailer is used to send email notifications, including password recovery and account verification.
  - **Database**
    - **Database Choice**: MongoDB Atlas is chosen for scalable and efficient storage of user data, course content, and progress tracking.
    - **Data Structure**: Data models are designed for students, instructors, and admin roles to manage unique functionalities, along with a course schema to handle multiple sections, student enrollments, and progress.

- **Workflow**
  - **Instructor Workflow**
    - Login -> Create/Edit/Delete Course -> Upload Videos via Cloudinary -> Manage Enrollments.
  - **Student Workflow**
    - Signup/Login -> Browse Courses -> Enroll -> Complete Course -> Obtain Certificate.
  - **Admin Workflow**
    - Login -> Monitor User and Course Activity -> Remove Content as Necessary.
- **Security Measures**
  - **JWT Tokens**: Used for secure session management and to protect sensitive routes.
  - **Password Hashing**: User passwords are securely hashed before storage to enhance account security.

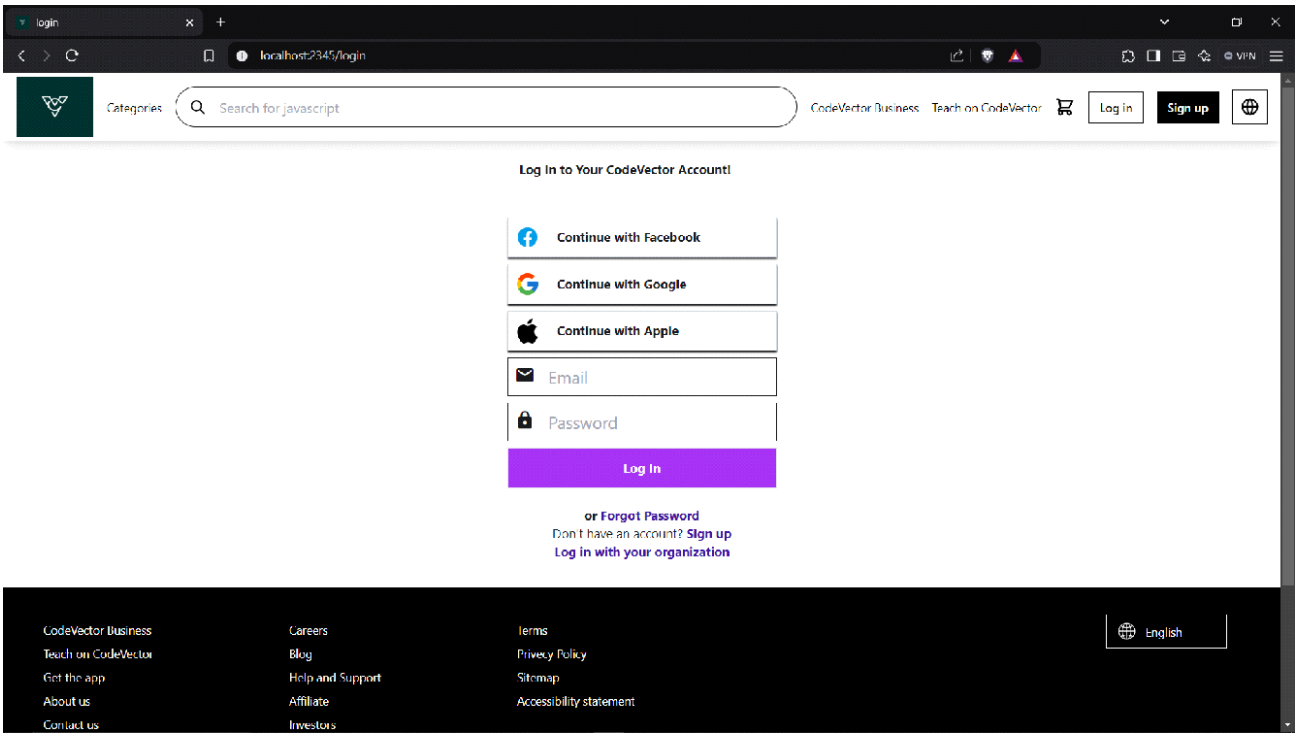
- **Future Scope**
  - Potential integration of a payment gateway for premium courses.
  - Addition of live class features using WebRTC or similar streaming technologies.
  - Enhanced analytics for instructors and admins to track engagement metrics.

Output Screenshots

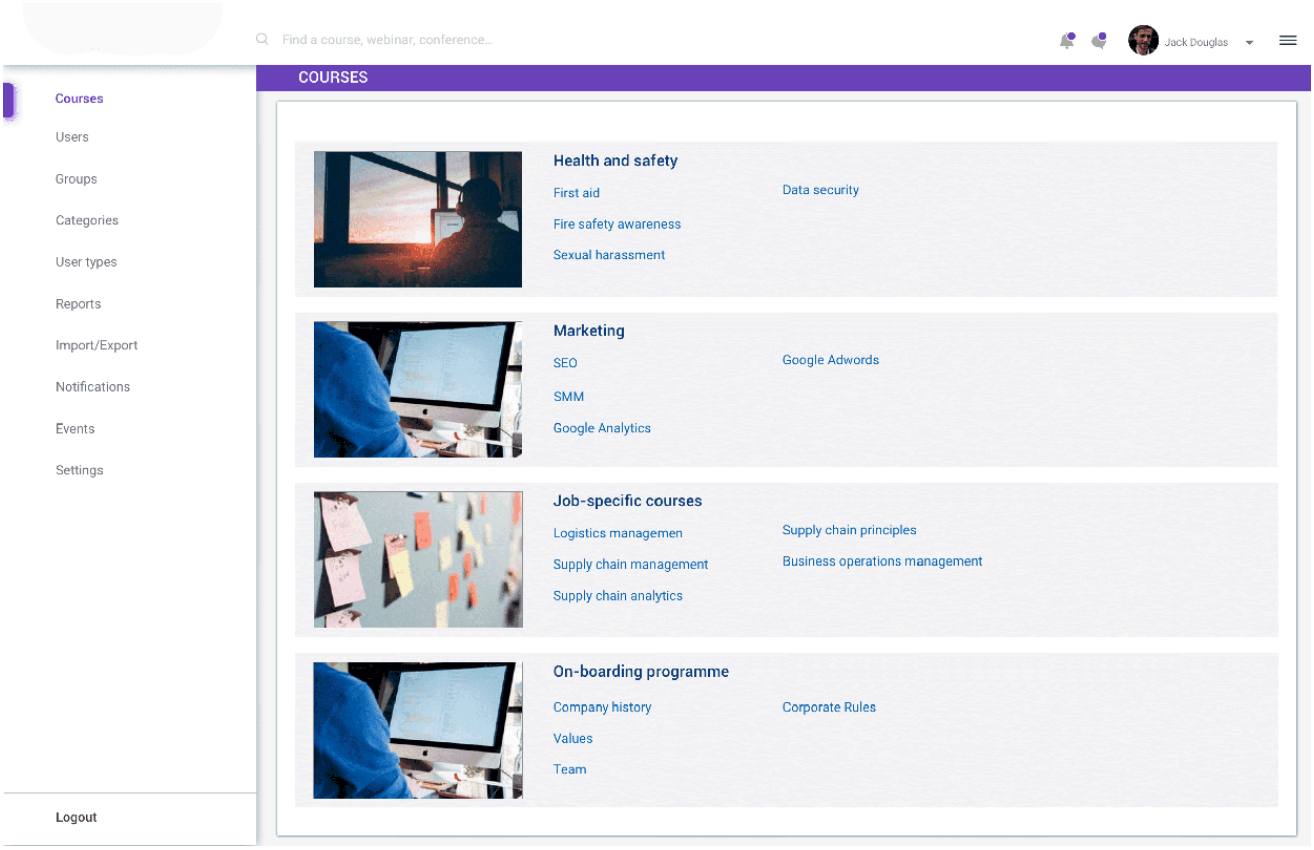
- **Landing Page**



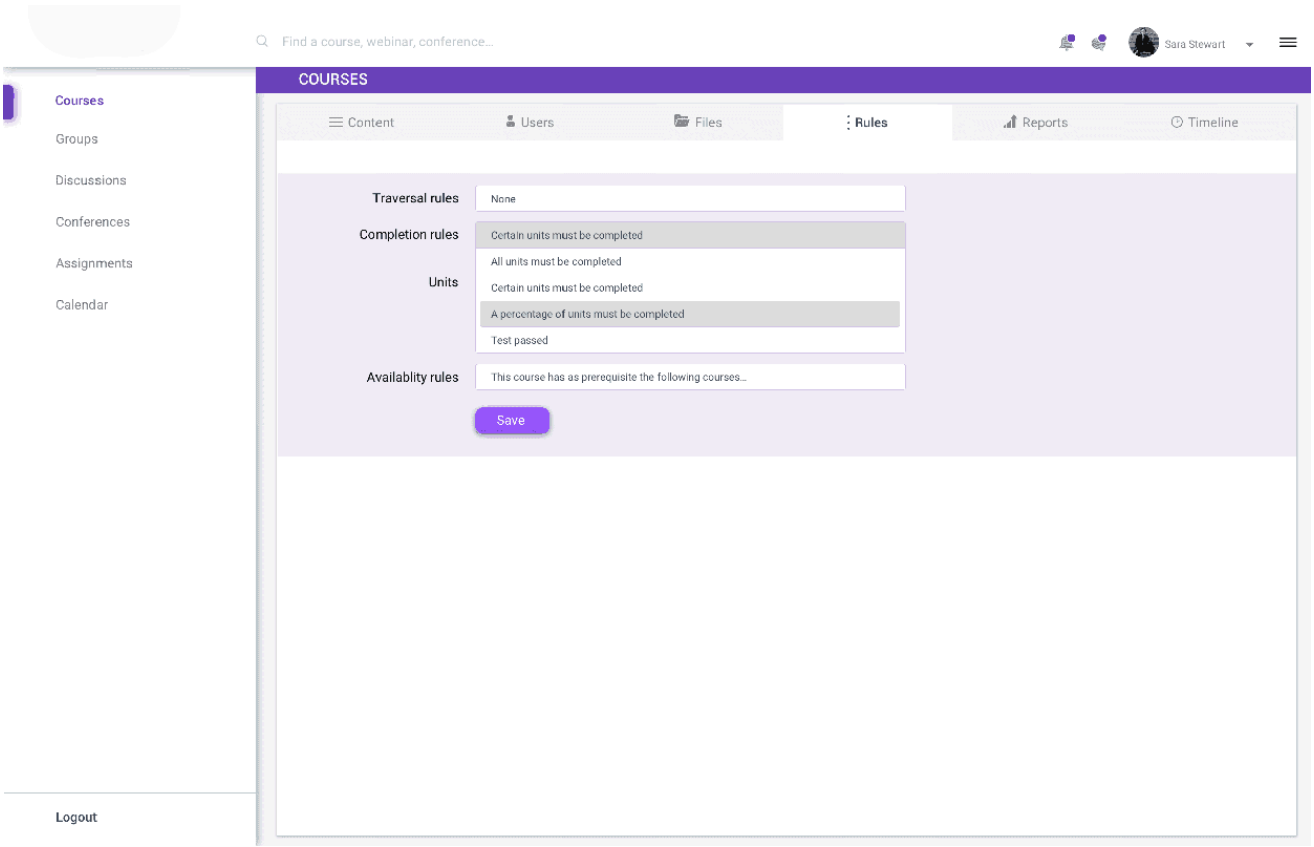
- **Instructor Login**



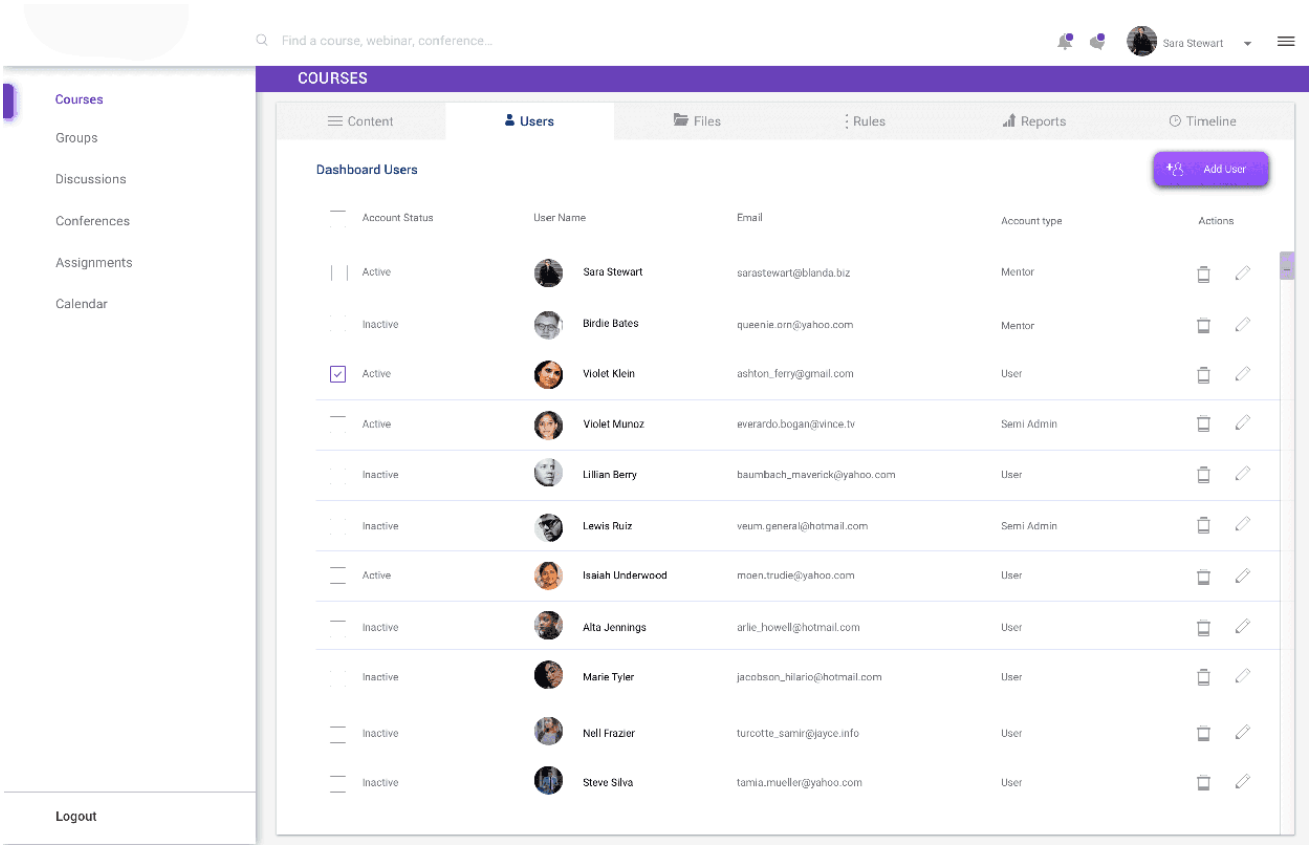
- **Instructor Dashboard**



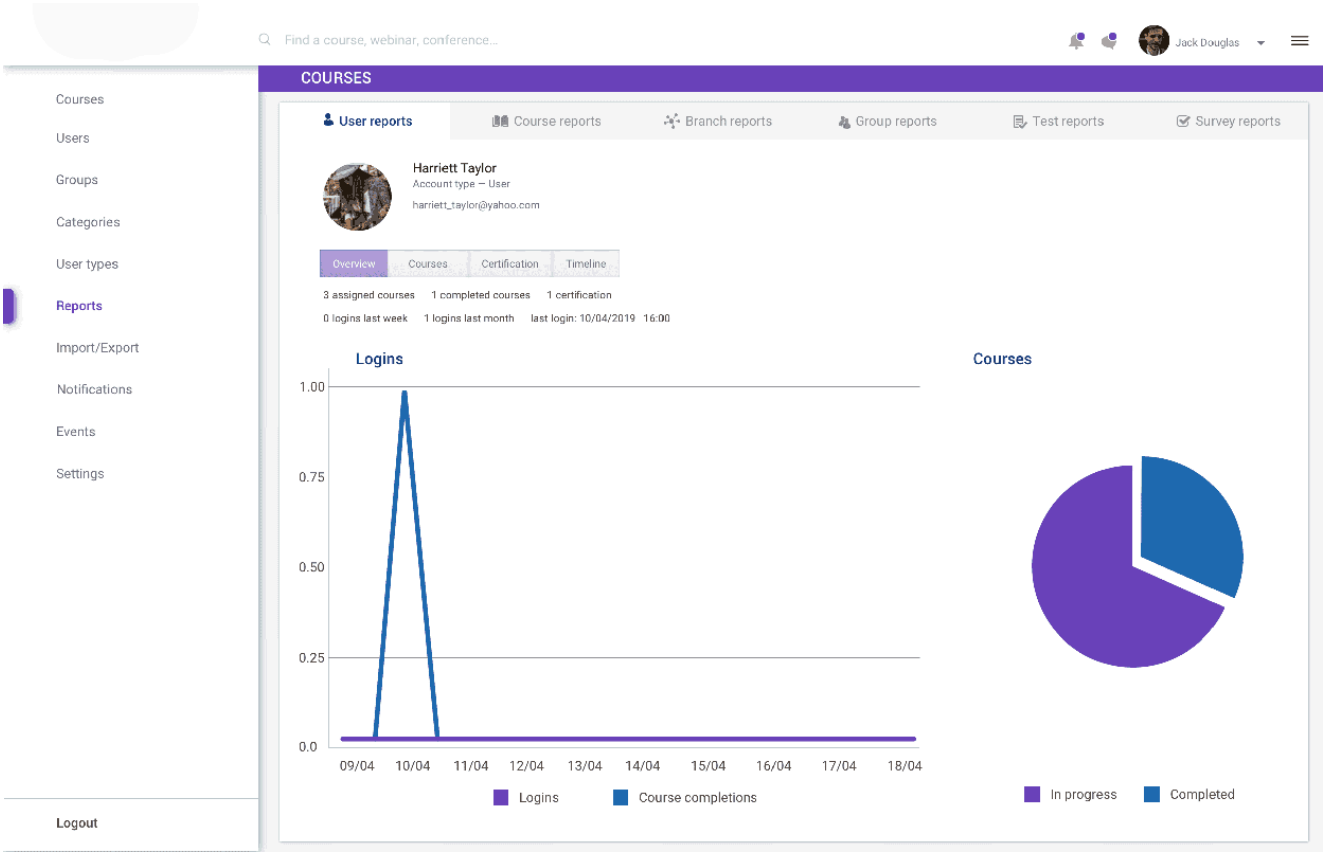
- **Course Creation**



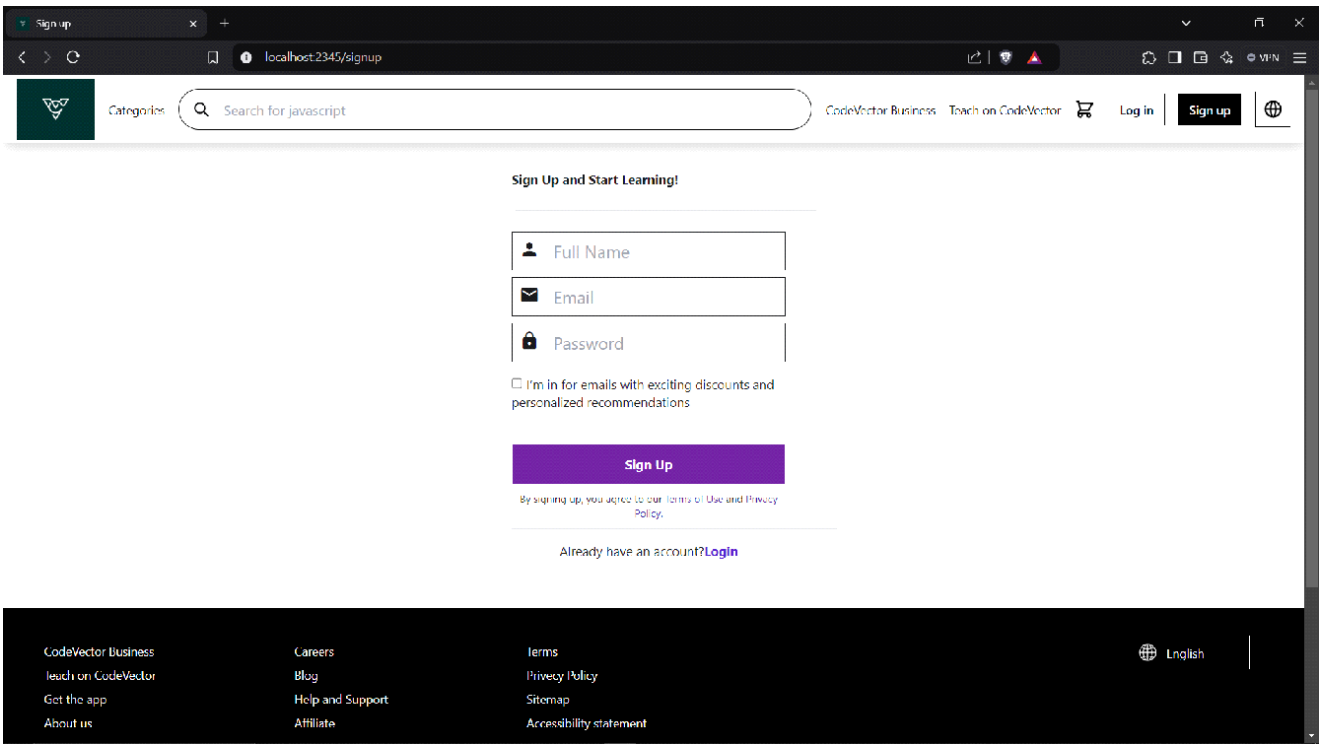
- **Enrolled Students**



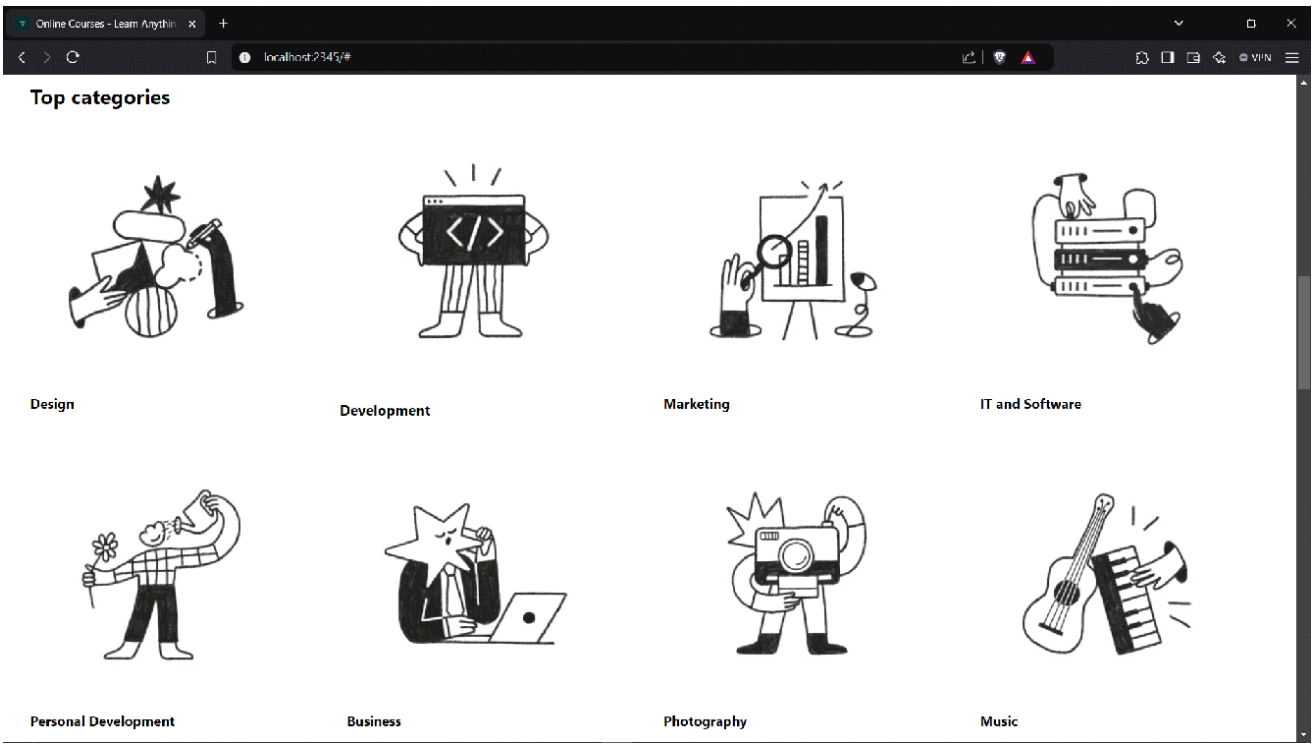
- **Profile Page**



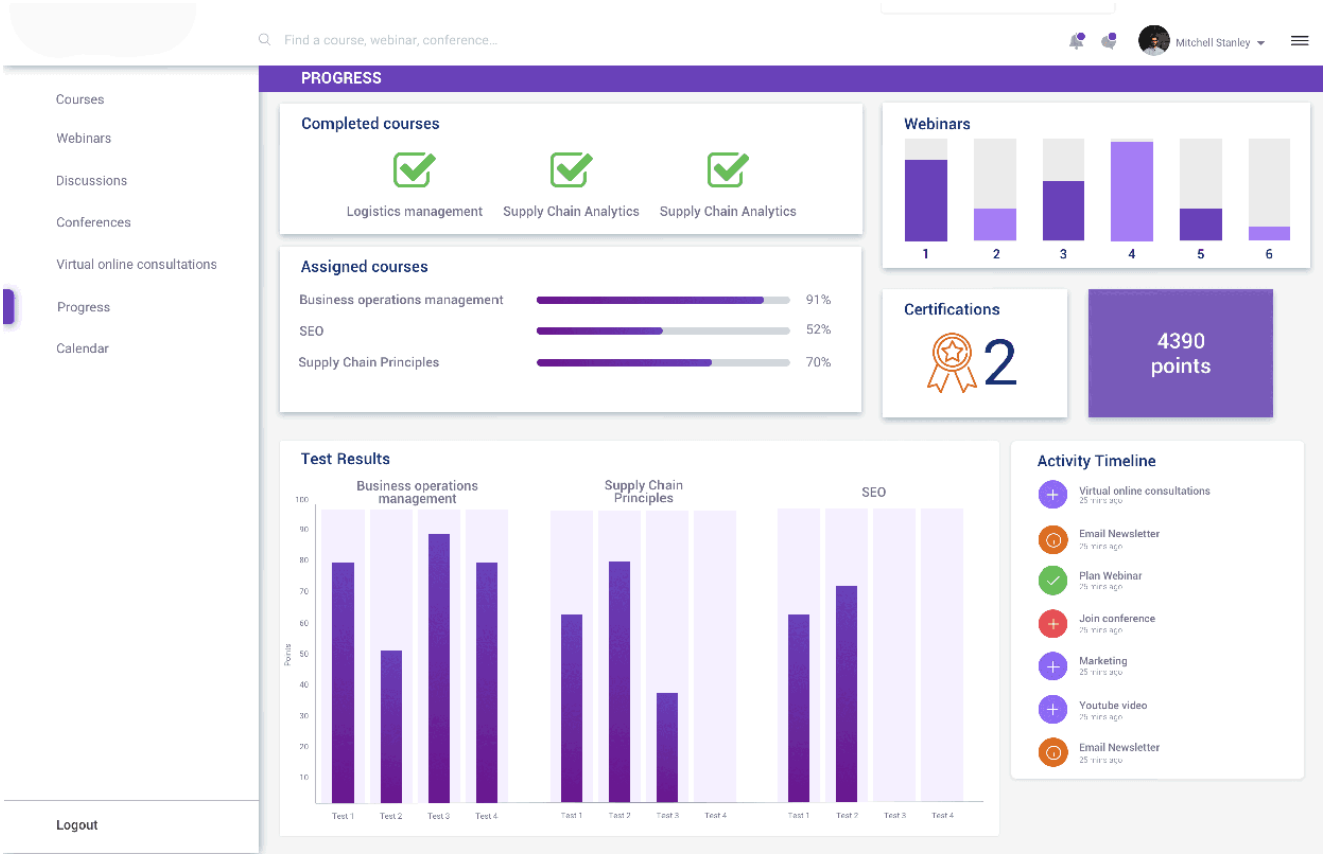
- **Student Login**



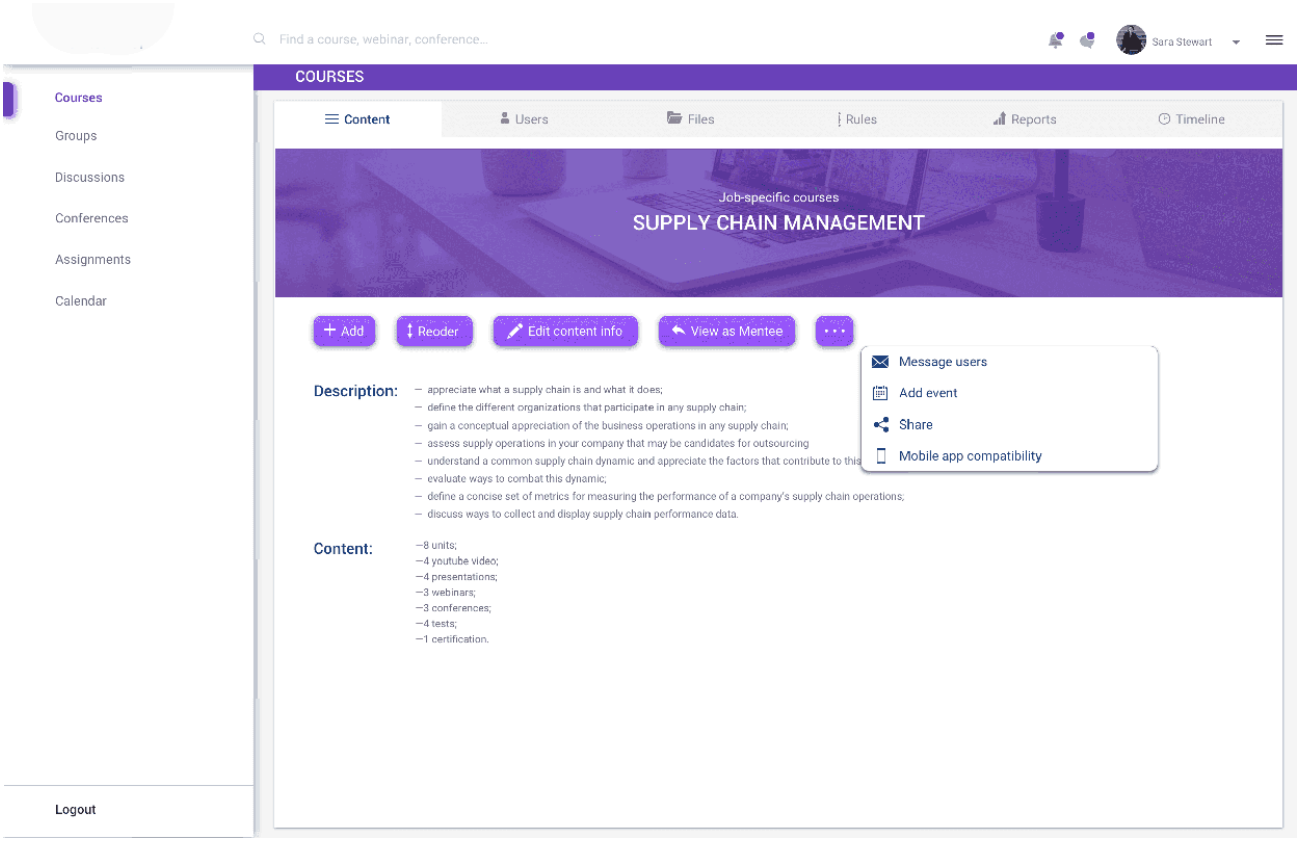
- **Student Dashboard – All courses tab**



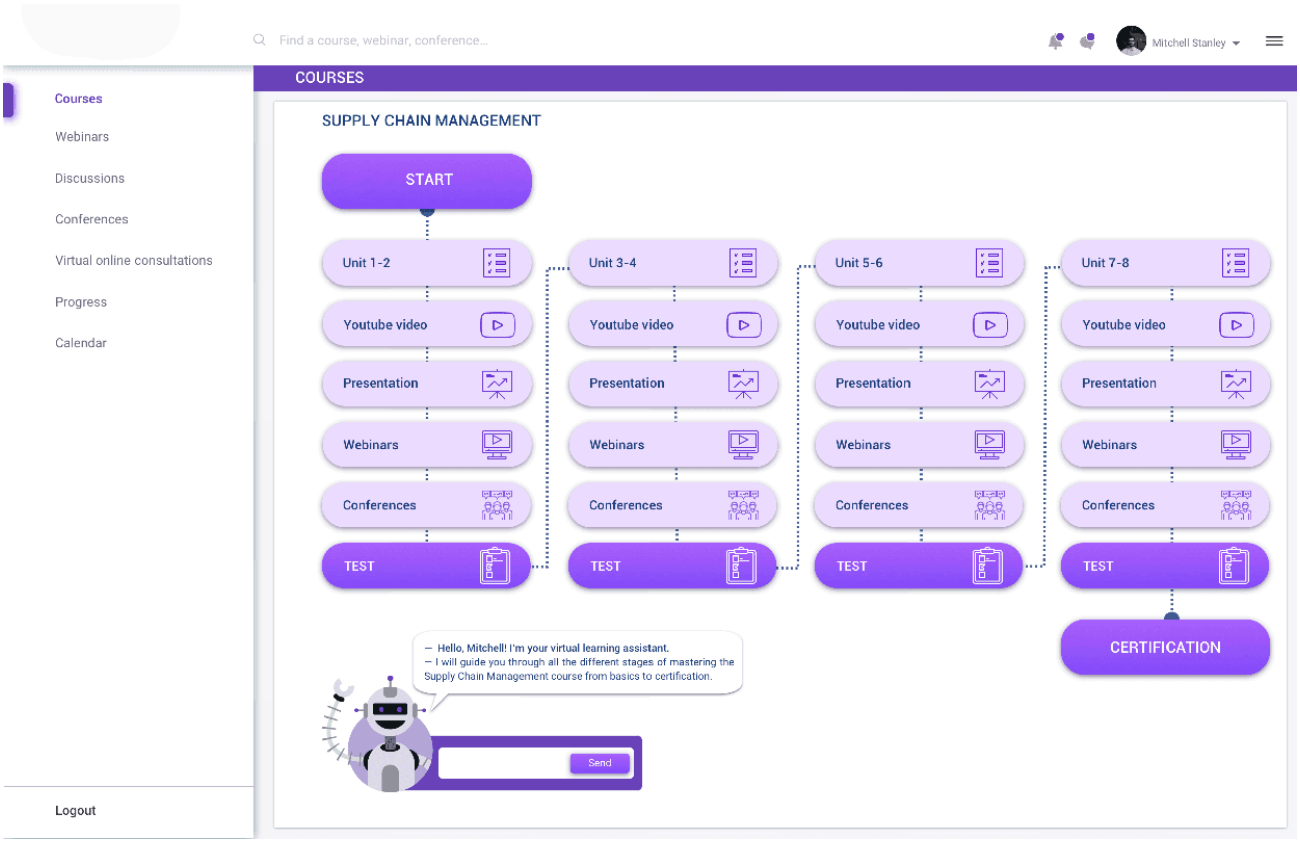
- **My courses Tab**



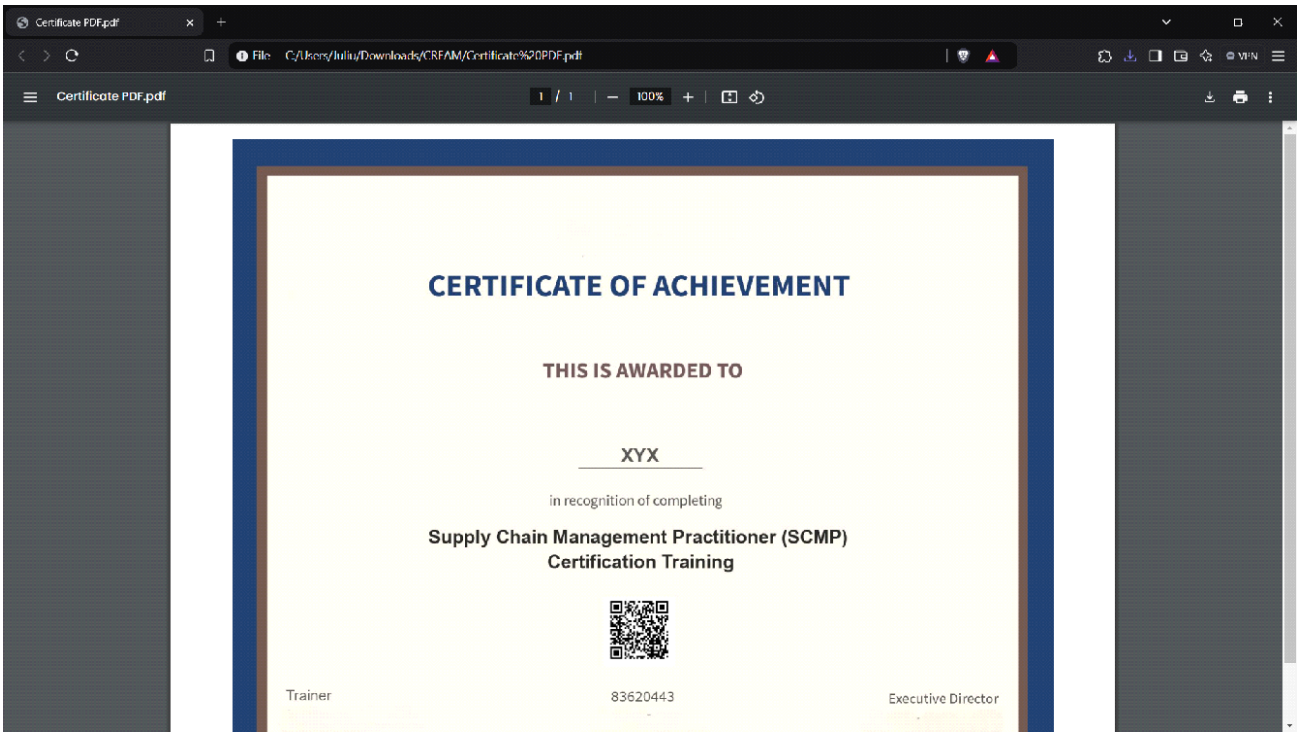
- **Course Details Page**



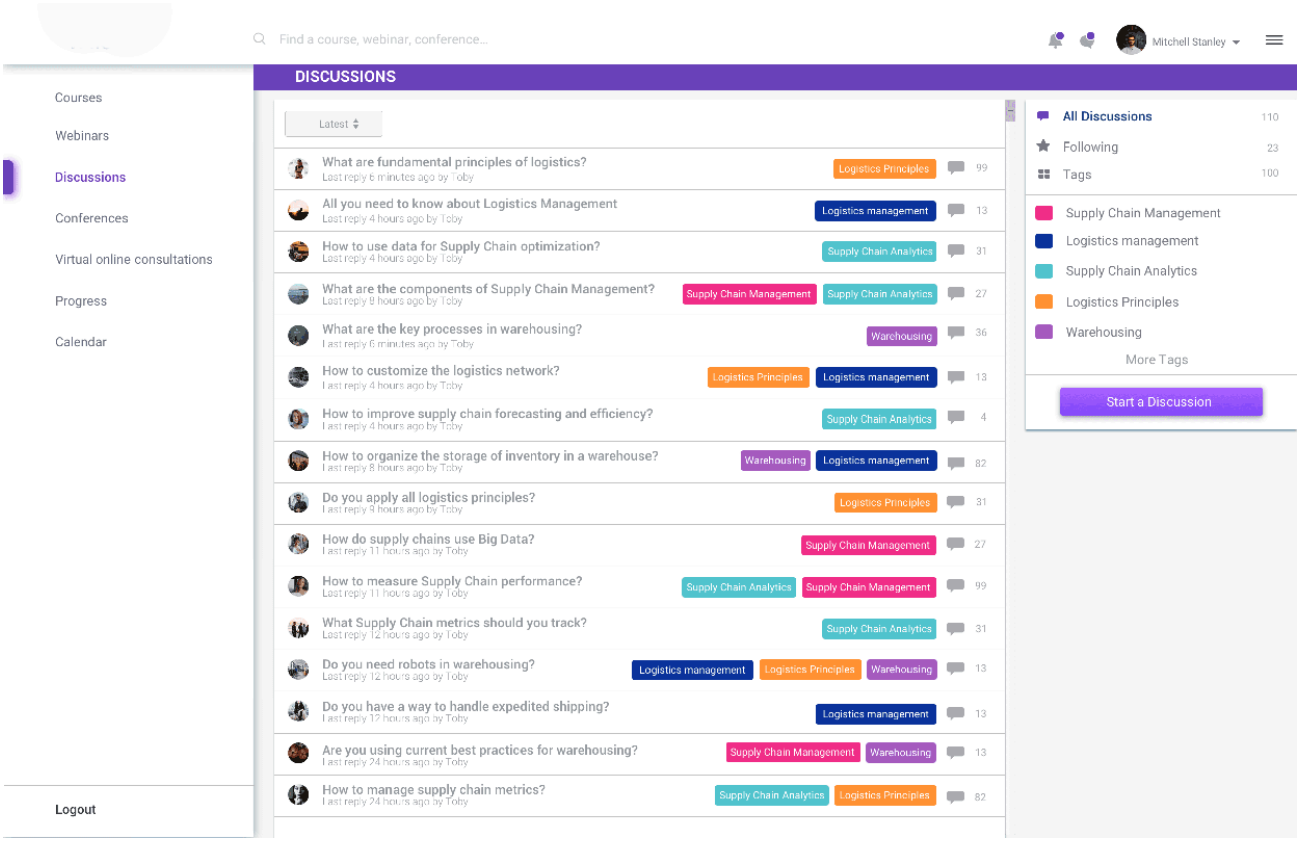
- **Course Accessed Page**



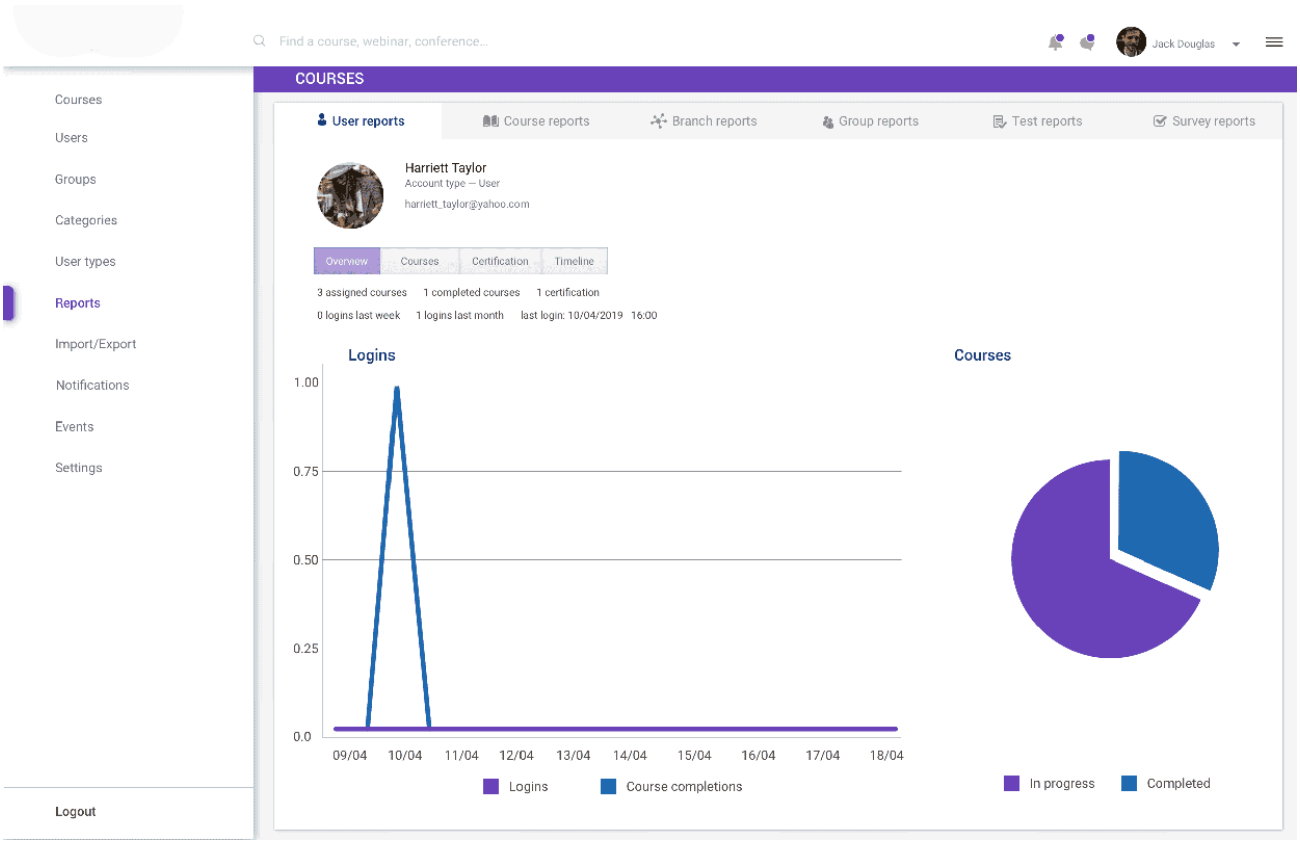
- **Certificate Download**



- **Discussion Page**



- **Student Profile Page**



- **Admin Dashboard**