Course Resource Library (CRL)

Final Project Documentation

Rishab Somraothi
Aaris Khan
Harshitha Yalama
Rahul Alladi

1. Introduction

The Course Resource Library (CRL) is a web-based software application designed to automate and streamline library management operations in educational institutions. It serves as a comprehensive platform for managing course-related resources and facilitating interactions between administrators (staff/faculty) and users (students/learners).

1.1 Purpose and Goal

- Create an efficient digital platform for managing educational resources
- Streamline the process of resource discovery and access for students
- Provide administrators with tools for effective resource management
- Automate routine library operations
- Enable secure user authentication and role-based access
- Facilitate course material distribution and management

1.2 Project Constraints

1. **Integration Complexity**:
    - Integration with existing learning management systems
    - Compatibility with current administrative software
    - Database integration challenges
2. **Resource Limitations**:

    - Budget constraints affecting development scope
    - Limited manpower for implementation
    - Hardware and infrastructure limitations
3. **Technical Constraints**:
    - Browser compatibility requirements
    - Security protocol implementation
    - Data protection compliance (GDPR, etc.)
4. **User Adoption**:
    - Training requirements for staff and users
    - Resistance to new system adoption
    - Learning curve considerations

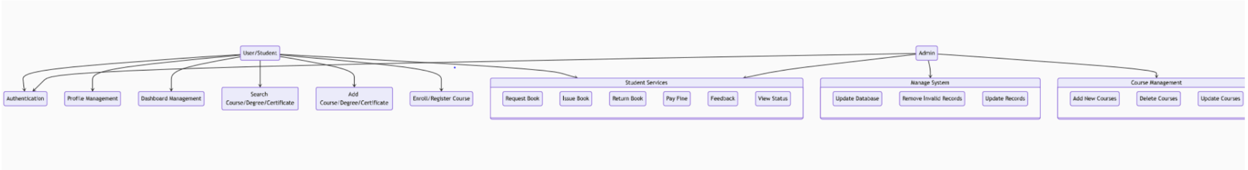
2. Requirement Glossary

1. **OPAC (Online Public Access Catalog)**: Digital interface for resource searching and browsing, including advanced search capabilities and filtering options.
2. **Cataloging**: Process of organizing and indexing library resources, including metadata management and classification.
3. **User Authentication**: Process of verifying user identity, including login credentials and access control.
4. **Resource Reservation**: System for requesting and holding materials, including waitlist management.
5. **Dashboard**: Personalized user interface showing relevant information, with different views for admin and regular users.

3. Use Case & Class Models

3.1 Use Case Diagram

The use case diagram represents the functional requirements and interactions of various actors (administrators and users) with the Course Resource Library system.



1. Profile Management:

Users can manage their personal profiles, including contact details, education, and professional experience. The system allows customization of privacy settings, notification preferences, and profile visibility, ensuring a personalized learning experience.

2. Dashboard Management:

The dashboard acts as a central hub showing users' learning progress, enrollments, and important updates. Users can customize their layout, track course progress, view deadlines, and receive notifications, all designed to enhance engagement and accessibility.

3. Course Search and Discovery:

Users can find courses through advanced filtering options based on title, subject, difficulty, and duration. The system includes smart recommendations based on user preferences and learning history, making it easy to discover relevant educational content.

4. Course/Degree/Certificate Management:

This module manages various educational offerings, including courses, degrees, and certificates. Users can access detailed information about curricula, learning outcomes, and accreditation, while maintaining wishlists and comparing options.

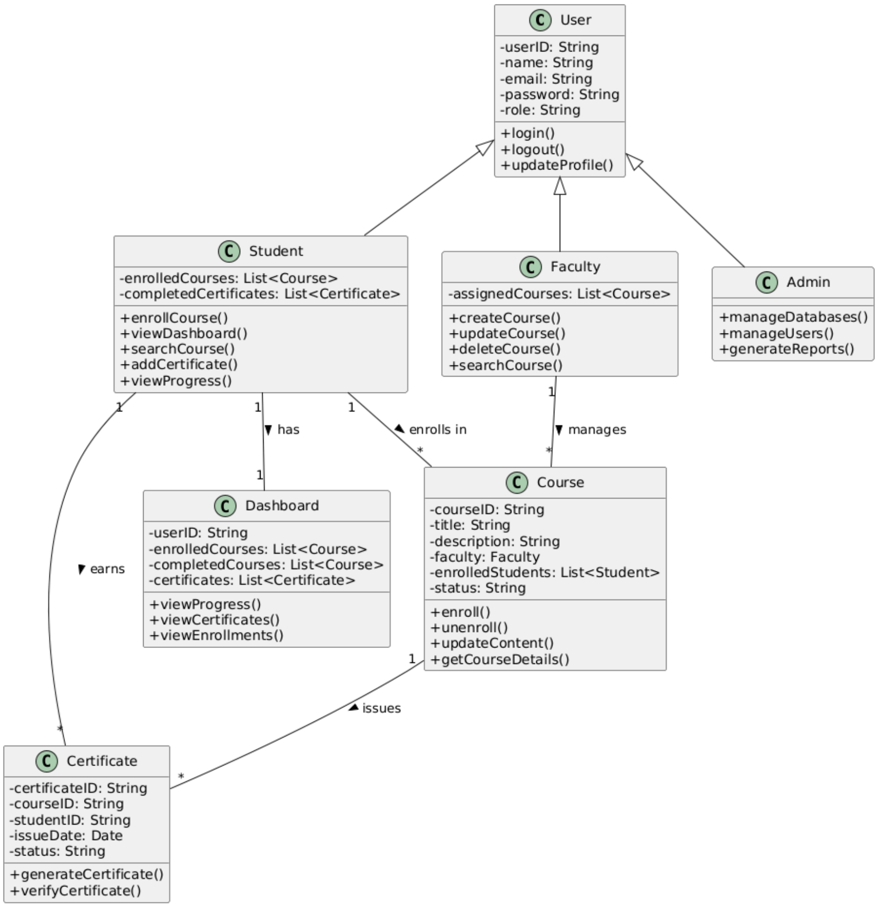
5. Enrollment and Registration:

The system streamlines course registration with efficient document submission and payment processing. It verifies eligibility, prerequisites, and manages waitlists while providing instant enrollment confirmation.

6. System Security and Access Control:

The system implements robust security measures including authentication, secure login, and role-based access control. It protects user data and maintains system integrity while ensuring appropriate access levels.

3.2 Class Diagram



The class diagram includes entities such as User, Administrator, Course, Resource, Enrollment, and Profile, along with their relationships (inheritance, association, and composition).

Here's an explanation of the class diagram components in that format:

1. User Management:

The User class serves as the base class for all users in the system, containing essential attributes like userID, name, email, and password. Users can perform basic operations such as login, logout, and profile updates. This ensures proper authentication and personalization of user experiences across the platform.

2. Student Functionality:

Students are special users who can interact with courses and certifications. They can enroll in courses, view their personalized dashboard, search through available courses, and track their progress. The system maintains lists of both enrolled courses and completed certificates, providing a comprehensive view of their academic journey.

3. Faculty Operations:

Faculty members have course management capabilities, represented by their relationship with courses. They can create new courses, update existing ones, delete courses when necessary, and search through course listings. The assignedCourses attribute maintains a list of courses under their supervision.

4. Administrative Control:

Administrators possess elevated system privileges for overall platform management. They can manage databases, handle user accounts, and generate system reports. This role ensures proper system maintenance and oversight of all operations.

5. Course Structure:

The Course class is central to the system, containing essential course information like courseID, title, and description. It manages the relationship between students and faculty, tracks enrolled students, and handles course status. Operations include enrollment, unenrollment, and content updates.

6. Dashboard Integration:

The Dashboard class serves as a visual representation of user progress and achievements. It maintains records of enrolled courses, completed courses, and earned certificates. Users can view their progress, certificates, and current enrollments through dedicated interface methods.

7. Certificate Management:

The Certificate class handles academic achievements, storing information like certificateID, courseID, and studentID. It includes functionality to generate new certificates and verify existing ones, maintaining the integrity of academic credentials.

8. Relationship Management:

The system uses various relationships between classes:

- Students "enroll in" courses and "have" dashboards

- Faculty "manages" courses

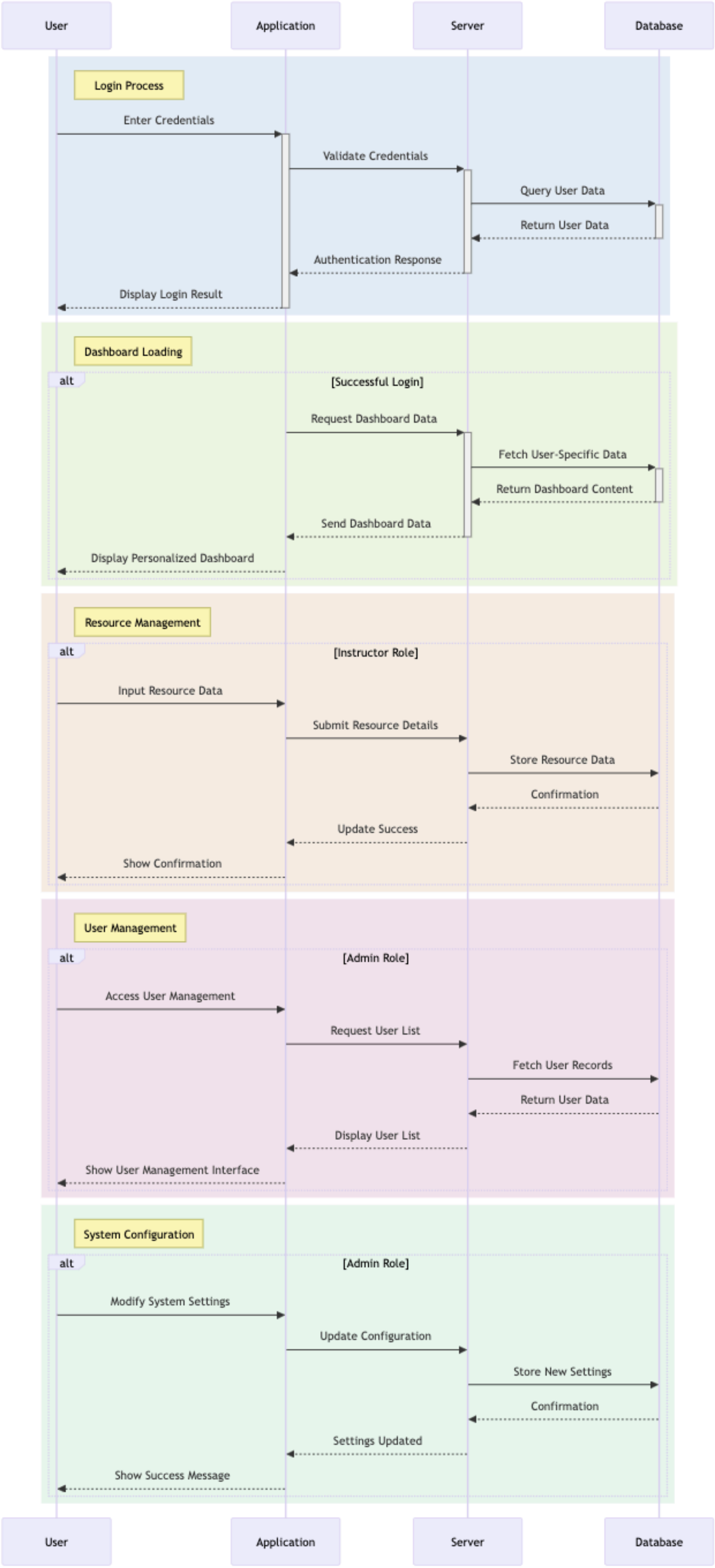
- Dashboards "issue" certificates

These relationships ensure proper data flow and functional organization within the system.

4. Design & Implementation Modules

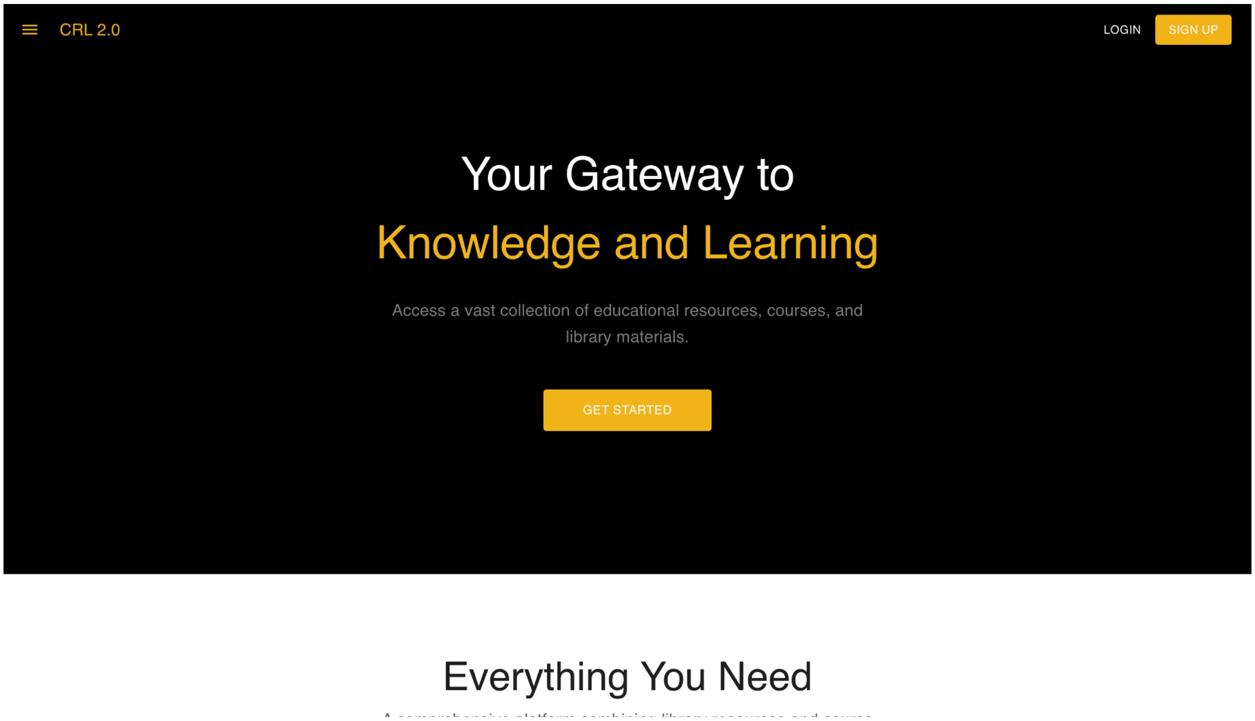
4.1 User Sequence Diagrams

1. **Login**: The login use case is crucial for providing secure access to the system. Users (both instructors and admins) need to authenticate themselves by entering their unique credentials (username and password).
2. **Personalized Dashboard**: After successful login, users are presented with a personalized dashboard tailored to their role within the system.
3. **Data Entry (Instructor)**: Instructors need to input resource data into the system, including details such as titles, descriptions, and metadata.
4. **Manage Users (Admin)**: Admins have the responsibility to manage user accounts within the system, including instructors and students.
5. **System Configuration (Admin)**: Admins have the authority to configure various system settings and parameters to customize the system according to institutional requirements.


User
Application
Server
Database
Login Process
Enter Credentials
Validate Credentials
Query User Data
Return User Data
Authentication Response
Display Login Result
Dashboard Loading
alt
[Successful Login]
Request Dashboard Data
Fetch User-Specific Data
Return Dashboard Content
Send Dashboard Data
Display Personalized Dashboard
Resource Management
alt
[Instructor Role]
Input Resource Data
Submit Resource Details
Store Resource Data
Confirmation
Update Success
Show Confirmation
User Management
alt
[Admin Role]
Access User Management
Request User List
Fetch User Records
Return User Data
Display User List
Show User Management Interface
System Configuration
alt
[Admin Role]
Modify System Settings
Update Configuration
Store New Settings
Confirmation
Settings Updated
Show Success Message
User
Application
Server
Database

5. Project Demo

The project demo showcases the various interfaces and functionalities of the Course Resource Library system, including the Get Started page, Login and Register page, Admin dashboard, Course management page, and User management page.

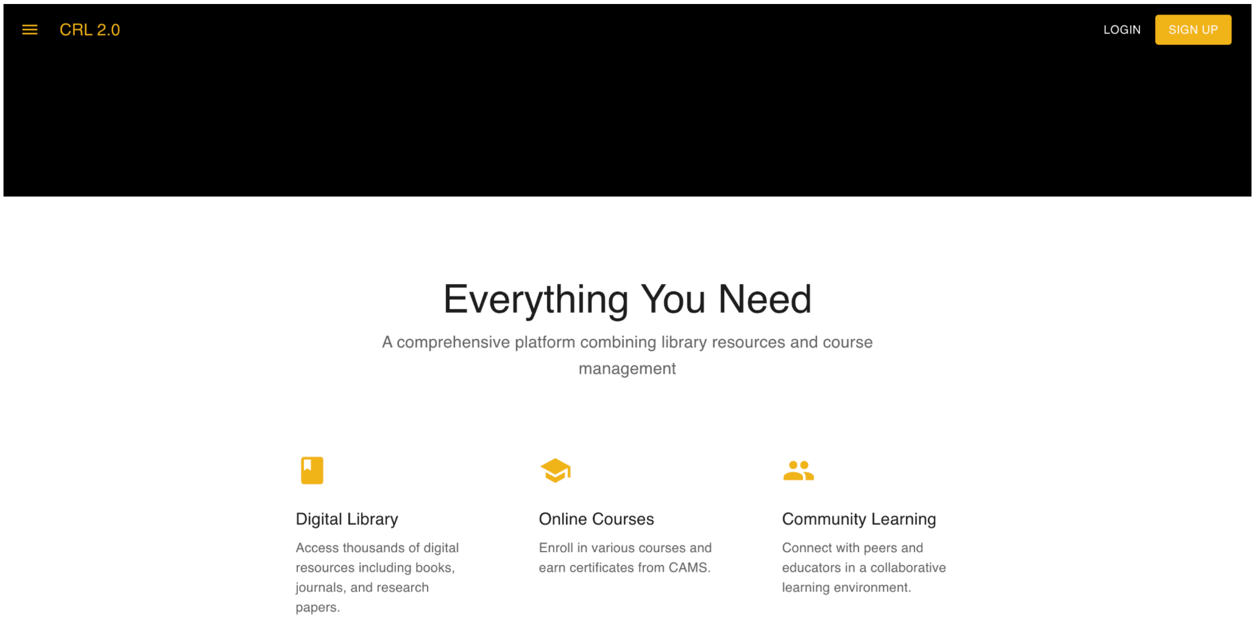


Your Gateway to
Knowledge and Learning

Access a vast collection of educational resources, courses, and library materials.

Everything You Need

A comprehensive platform combining library resources and course



Everything You Need

A comprehensive platform combining library resources and course management

Digital Library

Access thousands of digital resources including books, journals, and research papers.

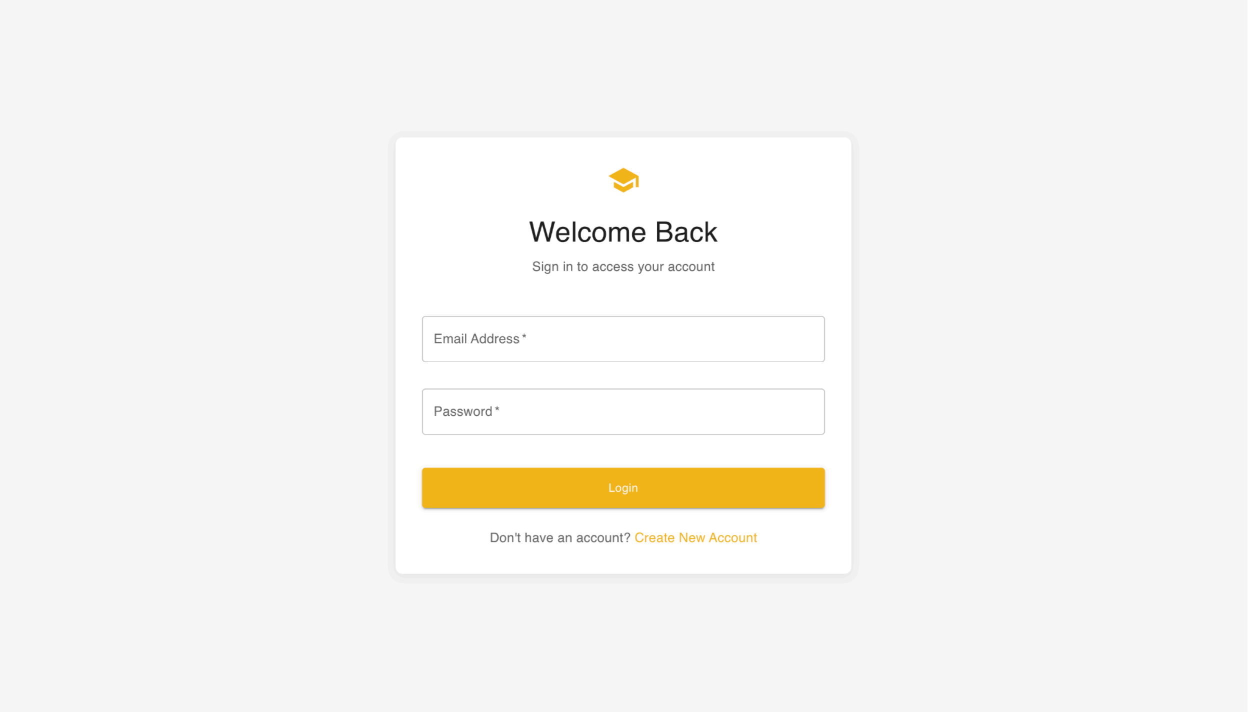
Online Courses

Enroll in various courses and earn certificates from CAMS.

Community Learning

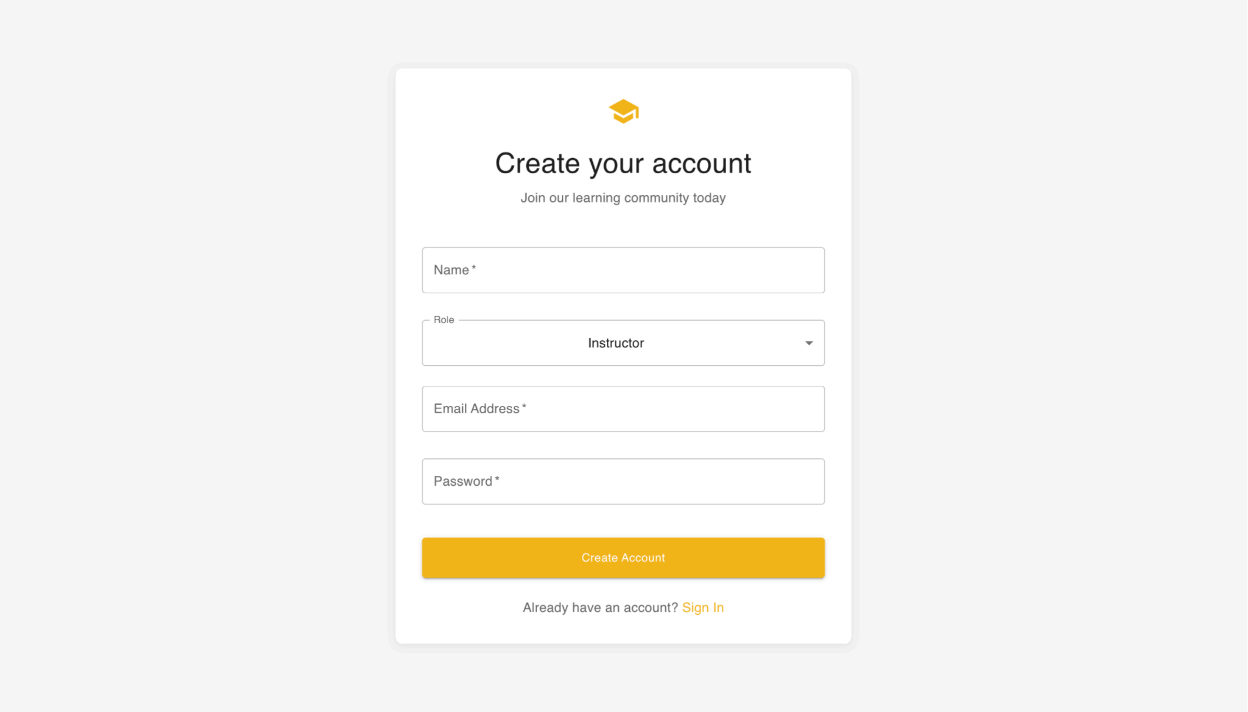
Connect with peers and educators in a collaborative learning environment.

Home page: The CRL2.0 home page which is simple shows the Login and Sign-up buttons , ensuring accessibility for users. It succinctly outlines the mission, functionality, and how users can get involved, providing a streamlined introduction to the platform.


Welcome Back

Sign in to access your account

Don't have an account? Create New Account

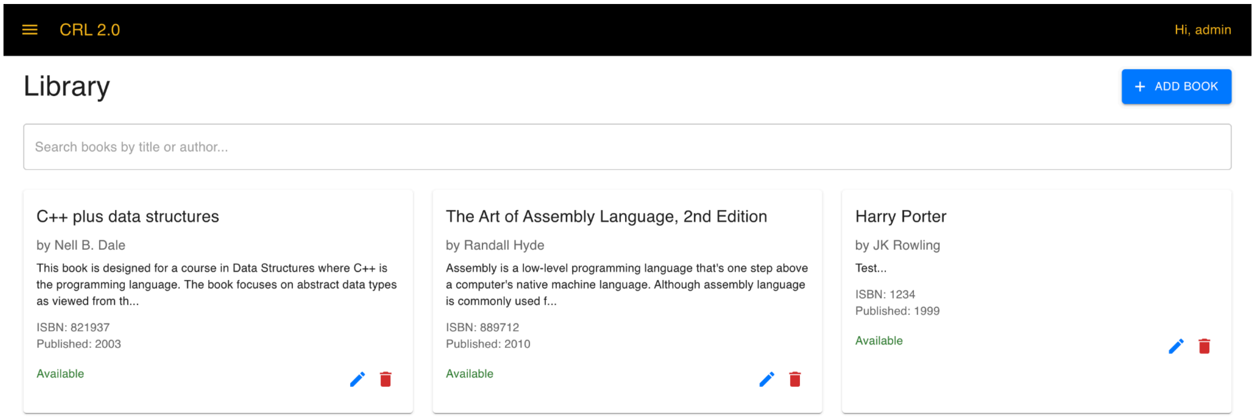

Create your account

Join our learning community today

Role: Instructor

Already have an account? Sign In

Login and Register Page: The Login and Register Page, users input their email and password for login or account creation. They must also select their role as student,instructor or admin. This ensures proper access control within the system.



Library

Search books by title or author...

C++ plus data structures

by Nell B. Dale

This book is designed for a course in Data Structures where C++ is the programming language. The book focuses on abstract data types as viewed from th...

ISBN: 821937
Published: 2003

Available

The Art of Assembly Language, 2nd Edition

by Randall Hyde

Assembly is a low-level programming language that's one step above a computer's native machine language. Although assembly language is commonly used f...

ISBN: 889712
Published: 2010

Available

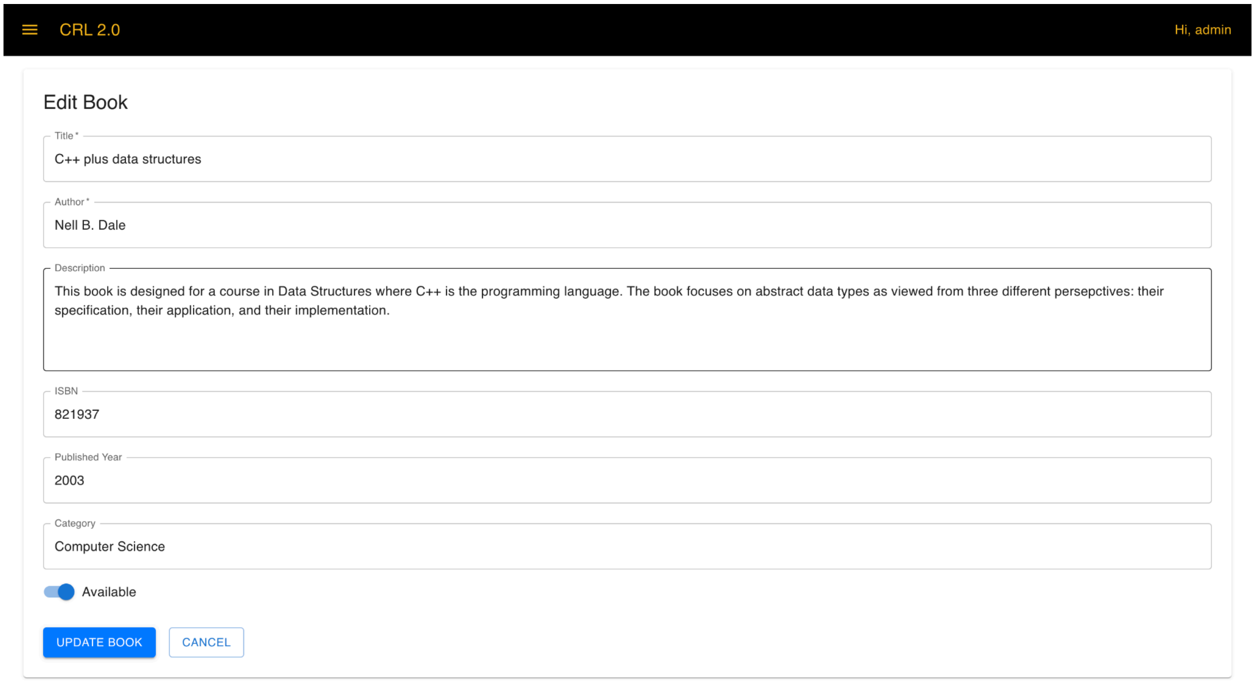
Harry Porter

by JK Rowling

Test...

ISBN: 1234
Published: 1999

Available



Edit Book

Title *: C++ plus data structures

Author *: Nell B. Dale

Description: This book is designed for a course in Data Structures where C++ is the programming language. The book focuses on abstract data types as viewed from three different persepctives: their specification, their application, and their implementation.

ISBN: 821937

Published Year: 2003

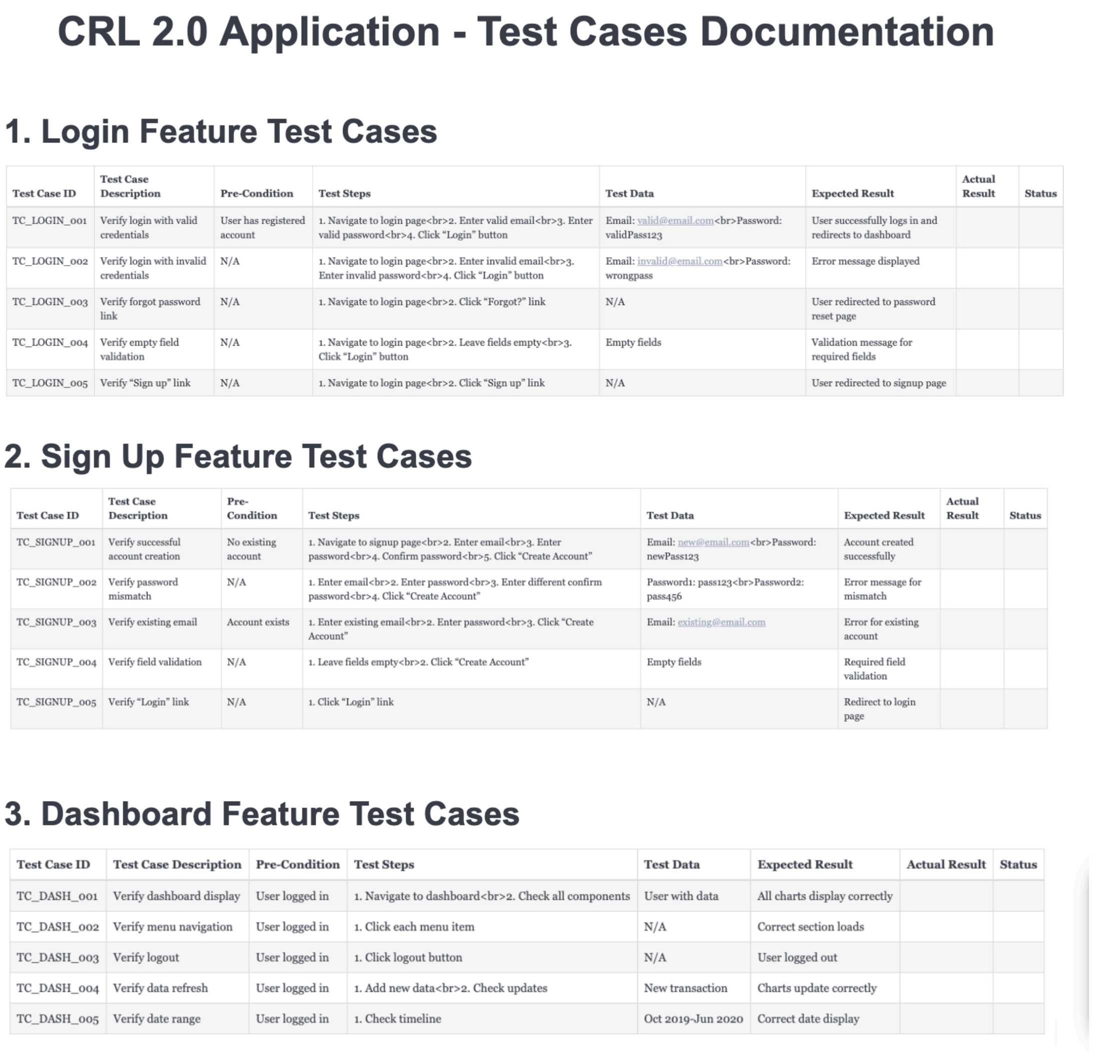
Category: Computer Science

● Available

Library Page: The Library page shows the list of available books. This can only be edited by admin. Students and Instructors will only be able to view this. This ensures proper access control within the system.

6. Testing Report

The testing report outlines the comprehensive testing procedures, methodologies, and results conducted on the CRL website, focusing on ensuring functionality, usability, security, and performance across various modules and features. The Cypress testing tool is utilized for automated testing.



CRL 2.0 Application - Test Cases Documentation

1. Login Feature Test Cases

Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result	Actual Result	Status
TC_LOGIN_001	Verify login with valid credentials	User has registered account	1. Navigate to login page 2. Enter valid email 3. Enter valid password 4. Click “Login” button	Email: valid@email.com Password: validPass123	User successfully logs in and redirects to dashboard		
TC_LOGIN_002	Verify login with invalid credentials	N/A	1. Navigate to login page 2. Enter invalid email 3. Enter invalid password 4. Click “Login” button	Email: invalid@email.com Password: wrongpass	Error message displayed		
TC_LOGIN_003	Verify forgot password link	N/A	1. Navigate to login page 2. Click “Forgot?” link	N/A	User redirected to password reset page		
TC_LOGIN_004	Verify empty field validation	N/A	1. Navigate to login page 2. Leave fields empty 3. Click “Login” button	Empty fields	Validation message for required fields		
TC_LOGIN_005	Verify “Sign up” link	N/A	1. Navigate to login page 2. Click “Sign up” link	N/A	User redirected to signup page		

2. Sign Up Feature Test Cases

Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result	Actual Result	Status
TC_SIGNUP_001	Verify successful account creation	No existing account	1. Navigate to signup page 2. Enter email 3. Enter password 4. Confirm password 5. Click “Create Account”	Email: new@email.com Password: newPass123	Account created successfully		
TC_SIGNUP_002	Verify password mismatch	N/A	1. Enter email 2. Enter password 3. Enter different confirm password 4. Click “Create Account”	Password1: pass123 Password2: pass456	Error message for mismatch		
TC_SIGNUP_003	Verify existing email	Account exists	1. Enter existing email 2. Enter password 3. Click “Create Account”	Email: existing@email.com	Error for existing account		
TC_SIGNUP_004	Verify field validation	N/A	1. Leave fields empty 2. Click “Create Account”	Empty fields	Required field validation		
TC_SIGNUP_005	Verify “Login” link	N/A	1. Click “Login” link	N/A	Redirect to login page		

3. Dashboard Feature Test Cases

Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result	Actual Result	Status
TC_DASH_001	Verify dashboard display	User logged in	1. Navigate to dashboard 2. Check all components	User with data	All charts display correctly		
TC_DASH_002	Verify menu navigation	User logged in	1. Click each menu item	N/A	Correct section loads		
TC_DASH_003	Verify logout	User logged in	1. Click logout button	N/A	User logged out		
TC_DASH_004	Verify data refresh	User logged in	1. Add new data 2. Check updates	New transaction	Charts update correctly		
TC_DASH_005	Verify date range	User logged in	1. Check timeline	Oct 2019-Jun 2020	Correct date display		

7. Tools and README

The development of the Course Resource Library (CRL) utilized the following tools and technologies:

- **Frontend**: React.js
- **Backend**: Node.js, Express.js
- **Database**: MongoDB
- **Testing**: Cypress
- **Deployment**: Netlify, Render
- **Version Control**: Git
- **Prototyping & Illustrations**: Figma

The README file provides detailed instructions for setting up and running the CRL application, including steps for installation, configuration, and deployment.

Github Project Repository Link

8. Project Risks

1. **Technical Complexity:** Integrating multiple technologies and frameworks such as React.js, Node.js, MongoDB, and API integration with Learning Management Systems (LMS) can pose challenges in terms of technical complexity. There may be difficulties in understanding the interactions between different components, leading to development delays or unexpected issues.

2. **Resource Constraints:** Limited availability of skilled developers, insufficient budget, or inadequate infrastructure may pose risks to the project timeline and deliverables. If the team lacks necessary

expertise or resources, it could lead to delays in development or compromises in the quality of the system.

3. **Compatibility Issues:** Ensuring compatibility with existing systems or software dependencies, such as Learning Management Systems (LMS) and student information systems, may present challenges. Incompatibilities could require additional development efforts or compromise the functionality of the Course Assessment Management System (CRL 2.0).

4. **Scope Creep:** Unclear or evolving requirements can lead to scope creep, where additional features or functionalities are requested beyond the initial project scope. This could result in delays, budget overruns, and dissatisfaction among stakeholders if not managed effectively.

5. **Performance Bottlenecks:** Inadequate scalability or optimization of the system may lead to performance issues, such as slow response times or system downtime during peak usage periods. Failure to address performance concerns could impact user experience and hinder the success of the Course Assessment Management System (CRL 2.0). These risks should be carefully monitored and mitigated throughout the development process to ensure the successful delivery of the project. Strategies such as regular communication, risk assessment, contingency planning, and proactive problem-solving can help mitigate these risks and ensure the project stays on track. Strategies for mitigating these risks include regular communication, risk assessment, contingency planning, and proactive problem-solving.

9. Functional and Non-Functional Requirements

The CRL project includes a comprehensive set of functional and non-functional requirements, covering areas such as user authentication, dashboard, data entry, collaboration features, user management, system configuration, user support and training, regulatory compliance, performance, usability, reliability, security, and scalability.

10. Good Programming Practices

The CRL project adheres to good programming practices, including modularization, error handling, design patterns, data security, testing, and automation.

11. Project Execution Results

The CRL project emphasizes good programming practices, including:

- Modularization
- Error handling
- Design patterns
- Data security
- Testing
- Automation

Project Deliverables

1. **System Requirements Specification (SRS) Document**
    - Outlines detailed requirements, functionalities, and constraints of the Course Assessment Management System
2. **Prototype Design**

   - Visual representation of the system's interface and user interactions
   - Showcases layout and features of the CAMS platform
3. **Login/Registration Implementation**
   - Creation of secure authentication mechanisms
   - User access control for instructors and administrators
4. **User Screens with Add/Edit Entries**
   - Development of user interfaces for data entry
   - Enables instructors to input assessment data efficiently
5. **Admin User Management**
   - Implementation of comprehensive user management functionalities
   - Allows administrators to add, remove, and delete user accounts
6. **Backend-Frontend Integration**
   - Connection between Node.js and MongoDB backend
   - Integration with React.js frontend
   - Ensures seamless data flow and system functionality
7. **Testing Implementation**
   - UI test automation using Cypress
   - Ensures system reliability and usability
   - Comprehensive testing framework implementation

Technical Stack:

- Frontend: React.js
- Backend: Node.js
- Database: MongoDB
- Testing: Cypress

12. Project Management

Implementation Timeline: 5 Sprints

Sprint 1: Foundation (2 weeks)

- System architecture design
- Database schema development
- Basic user authentication setup
- Environment configuration
- Project repository setup

Sprint 2: Core Features (2 weeks)

- User registration and login system
- Role-based access control
- Admin dashboard development
- Basic course management features
- Database integration

Sprint 3: Resource Management (2 weeks)

- Course resource upload functionality
- Resource categorization system
- Search and filter capabilities
- Student enrollment system
- Assignment creation tools

Sprint 4: Assessment & Analytics (2 weeks)

- Grade management system
- Progress tracking features
- Performance analytics
- Reporting system
- Data visualization tools

Sprint 5: Testing & Deployment (2 weeks)

- Comprehensive testing
    - Unit testing
    - Integration testing
    - User acceptance testing
-
- Bug fixes and optimizations
- Documentation
- Production deployment
- User training

Key Milestones: ✓ Sprint Reviews: End of each sprint ✓ User Testing: Throughout Sprint 4-5 ✓ Final Deployment: End of Sprint 5