

Software Requirements and Specification(SRS)

For

Acadonline:content sharing educational website

Group: 13

Index:

1. Introduction:	3
1.1 Purpose	3
1.2 Intended audience	3
1.3 Scope	3
1.4 Goals/objectives:	3
2. Overall Description:	5
2.1 Product perspective	5
2.2 Product Functions	5
2.3 User class and Characteristics	5
3. Specific Requirements:	6
3.1 External Interfaces	6
3.2. Functional Requirements:	7
4. System Features:	8
4.1. Use-cases:	8
4.2. Use-case description:	9
5. Non-Functional Requirements:	19
6. Future Scope:	20

1. Introduction:

1.1 Purpose

The purpose of our website is to provide a platform for educators and students and to enhance their learning experience. The primary aim of an education content-sharing website is to democratize access to high-quality educational resources, making them available to anyone with an internet connection.

1.2 Intended audience

This SRS document is meant for developers to analyze user requirements and implement functionalities and also for the stake holders to review the system requirements and for testers to specifically test this requirement.

Apart from this, it is also meant for students and educators and general public who are potential users of this system.

1.3 Scope

This system's scope is to encourage people's learning experience and collaboration and knowledge sharing among students and educators. Also, Educators can get feedback based on their content rating. This system will provide free education to anyone with only an internet connection.

1.4 Goals/objectives:

User Authentication: The system should allow users to authenticate themselves by providing their credentials.

User Roles: The system should have different user roles with different access levels. Like, admin should have additional rights of removing user and add courses other than normal users.

Course content access: The system should be able to provide all content uploaded on website to users, and also provides quality of content to users like rating of the content.

User Profile: User should be able to view their user profile and make changes to it.

User Interface: The system should have an intuitive and user-friendly interface that allows users to easily navigate and find the information they need.

Accessibility: The website should be available to all the users which only have a internet connection

Scalability: The system should be scalable to accommodate growing number of users on website.

2. Overall Description:

2.1 Product perspective

Our website is a new, self-contained product that aims to provide a platform for educators and learners to create and share high-quality educational content. It is not part of an existing product family nor a replacement for any existing systems.

Overall, the content sharing educational website will be a self-contained platform designed to provide high-quality educational content and facilitate collaboration and communication among users. It will be a valuable resource for learners, educators, and content creators, and will provide opportunities for revenue growth and user engagement.

2.2 Product Functions

- User will be able to log in and Signup.
- User will be able to view all courses.
- User will be able to see course content of all courses.
- Users will be able to manage their own content.
- Users will be able to discuss their doubts in the Q&A section.
- Admin will be able to manage courses on the website.
- Admin will be able to manage users on the website.

→ Detailed functions will be discussed in section 3.

2.3 User class and Characteristics

User: Anyone who has the desire to learn new things and share their experience with other people.

3. Specific Requirements:

3.1 External Interfaces

VS code:

VS code helps in using extensions that enhance our development experience. It also allows us to integrate with git so we can directly manage our version control from the editor.

Github:

A version control system to assist us in maintaining the project while working with the full team.

EJS:(Embedded javascript)

EJS is a simple and versatile templating language that allows for easy integration with JavaScript frameworks.

MongoDB:

MongoDB offers high scalability and flexibility in data modeling, making it ideal for handling large amounts of unstructured data in modern web applications.

Bootstrap:

Bootstrap provides a comprehensive set of pre-designed UI components and responsive layout systems, making it easy to create visually appealing and mobile-friendly web applications with minimal effort.

Node.js:

Node.js offers high performance and scalability in server-side JavaScript execution, making it ideal for building fast and scalable network applications with event-driven architecture.

3.2. Functional Requirements:

Login/Signup:

- Users can log in using userID and Password.
- User should be able to reset the password and log in again.
- User will be able to register on the system with some user details.
- User will be able to be logged in until 7 days until he/she manually logout.
- User will be able to forget the password and log in again with a new password.

Course:

- User should be able to view all the courses available on the website.
- User should be able to search courses on the website.
- Admin should be able to add courses.
- Admin will be able to enter a course name, description about the course, and photo related to that course.
- Admin should be able to edit and remove the course.

Course-Content:

- User should be able to see all content available for a particular course.
- Users should be able to see course content sorted by their rating.
- Any user should be able to add content for a course.
- Users should be able to add their content video content and documents to the website.
- User should be able to manage(edit,delete) their own content.
- User should be able to give a rating to any particular course content, and also able to clear rating and change rating.
- User should be able to manage course content from his/her user profile.

Q&A section:

- User should be able to see all questions and related replies to that particular question.
- User should be able to add questions to any particular course.
- User should be able to add a reply to any question in the Q&A section.
- User should be able to manage(edit/delete) his/her question.
- User should be able to delete his/her reply to a question.

4. System Features:

4.1. Use-cases:

Actors: User and Admin

- User:
 1. Login/Register to the System
 2. Search the course
 3. View the courses
 4. View the course content
 5. Give Ratings on course content
 6. Put Questions on course
 7. Add a reply to a Question on course
 8. Login/Register to the System
 9. Add his/her course content
 10. Delete his/her course content
 11. Edit his/her course content.
- Admin:
 1. log in to the system
 2. Remove User from System
 3. Course Management, Like Adding/Removing Course

4.2. Use-case description:

1. Register

Actors: User, System

Pre-condition: The user has entered the correct URL and should be on the right page.

Goal: The user should be able to register into the system

Basic Flow:

1. User will be asked to enter username, email, and password and confirm the password and institute name.
2. User will enter all details on the register page.
3. User clicks the “register” button.
4. System will verify details, register the user, and redirect to the login page.

Alternate Flow:

- 4a. If the password doesn't match with criteria, the user will receive error
- 4b. If the password and confirm password don't match, the User will receive an error.

Postcondition: The user will be registered into the system and redirected to the login page

2. Login:

Actors: User, system, admin

Pre-condition: The user has entered the right URL and should be on the right page.

Goal: The user should be able to login into the system

Basic Flow:

1. User will be asked to enter his username or email and password for login.
2. User will enter username, email, and password into the login page and click “login.”
3. The system will verify the details, and the user will be logged in.

Alternate Flows:

- 3a If user credentials are wrong, then the system will give the error “Invalid Credentials.”

Postcondition:

The user will be logged into the system.

3. Search course

Actors: user

PreCondition: The user is logged in to the system

Goal: Students will be able to see various courses on the system.

Basic Flow:

1. User will select the search bar for searching courses.
2. User will enter the course name for which he/she wants to see the content.
3. System will display the course that the user has entered.

Alternative Flow:

- 3a. User will enter the name of the course that is not available then the system will show the message no course found.

PostCondition: The user will be able to see the course he/she has searched

4. View course:

Actors: user

PreCondition: The user is logged in to the system

Goal: Students can view the course

Basic Flow:

1. User will select “view all courses” or search particular course.
2. System should display details like course name and course description about that course.

Alternative Flow: None

PostCondition: The user will be able to see the course.

5. View course content

Actors: user

PreCondition: The user is logged in to the system

Goal: Students can view the course content

Basic Flow:

1. Users will search for the course they want to see the content.
2. System will display course content available for the searched course.
3. Users will select the course content they want to see.

Alternative Flow: None

PostCondition: The user will be able to see the content of the course.

6. Add Course Content

Actors: User

Precondition: The user should be logged in and have proper permissions to add the content

Goal: Educators should be able to add the content.

Basic Flow:

1. User will select the “Add content” option.
2. User will be prompted to fill out the content addition form.
3. User will enter the details about course content like content description, File upload link, etc.
4. User will click the “submit” button.
6. System will create new course content for a particular course in the database.

Alternate Flows:

5a. If the user chooses to cancel, then the system doesn't add the new course Content.

Postcondition: Course content will be displayed on the page added by the user.

7. Remove Course Content

Actors: User

Precondition: The user should be logged in and have proper permissions to remove the content

Goal: Educators should be able to remove the course content.

Basic Flow:

1. User will select the “Remove content” option.
2. System will ask the user, “Are you sure to remove this content?”.
3. User will choose “Yes.”
4. The system will remove the course content for a particular course from the database

Alternate Flows:

5a. If the user chooses “No,” then the system doesn't Remove the course content.

Post-Condition: The user will be able to remove course content from the course.

8. Q&A section:

Actors: User

PreCondition: The user must be logged in to the system.

Goal: students' questions can be solved

Basic Flow:

1. User will be prompted to the Q&A section page.
2. If a student wishes to ask a question, then click the “ask question” option.
3. System will prompt students to enter their question details.
4. User will enter a question description in the forum and click the “post” button.
5. System will post this question on a Q&A section of that particular course and be available to everyone to answer.
6. If a user wants to answer a particular doubt, click on reply, and then he/she will enter an answer to that doubt.

Alternative Flow: None

PostCondition: User will be able to ask questions and get answers to their Doubts.

9. Rate the courses

Actors: User

PreCondition: The user is logged in to the system

Goal: The user will be able to rate the course based on its quality

Basic Flow:

1. User will have the option of rating us on the content page.
2. User will give a rating based on his relevance to course content.
3. User has to click on 'submit your rating'.
4. System will add the user's rating and also update the current rating.

Alternative Flow: None

PostCondition: The system should count users for every star, then show the average for that course.

10. Remove users

Actors: system database, admin

PreCondition: The admin should be logged in.

Goal: Inappropriate person can be removed

Basic Flow:

1. Admin will able to see the list of all users
2. If an admin wants to remove any person, he has to click the “delete” button located beside the user name.
3. Admin will be asked, “Are you sure to delete the user?”.
4. Admin will select “Yes”.
5. System will delete the user from the database.

Alternative Flow:

- 4a. If the admin will select “No”, the user will not be removed from the database.

PostCondition: The user will be removed from the database.

11. Add courses

Actors: admin

PreCondition: The admin should be logged in.

Goal: The admin should be able to add courses to the system.

Basic Flow:

1. Admin will select the “Add course” option.
2. Admin will be prompted to fill out the content addition form.
3. Admin will enter the details about the course like course name, course description and course photo, etc.
4. The System will verify the details and add a new course into the database.

Alternative Flow:

- 4a. If any of the course content addition form fields is empty so, the user will be guided to remove that.

PostCondition: New course will be added to the system.

12. Remove courses

Actors: admin

PreCondition: The admin should be logged in.

Goal: To remove the course

Basic Flow:

1. Admin will select the “Remove course” option.
2. System will ask the user, “Are you sure to remove this course?”.
3. Admin will choose “Yes.”
4. The system will remove the course from the database.

Alternate flows:

- 4a. If the user chooses “No,” then the system doesn't Remove the course.

PostCondition: The course deleted by the admin is removed from the system.

5. Non-Functional Requirements:

1. **Scalability:** The website should be able to handle community growth with an increasing number of students and educators.
2. **Security:** The personal information of students and Educators will be stored securely on the website.
3. **Ease of Use:** The Website should have an attractive UI/UX design along with some good functions to provide students and educators with a great Experience.
4. **Correctness:** The system should be able to handle different kinds of input and give the proper response.
5. **Performance:** The Website should load quickly even with a lot of traffic.
6. **Maintainability:** The addition or deletion of content should be immediately visible to the database, even if there is a server crash.
7. **Responsiveness:** The website should support both Laptop screens and mobile screens at the same time.

6. Future Scope:

- System should store video and documents in server and provide video and document rather than link.
- System should do AI integration and recommend courses based on user search history.
- System should use AI and also implements suggesting course content of the user which have high rating in all of his/her content.
- System should implement user feedback system where user can enter their reply in text and content adder can get idea about what changes he has to made it in his/her document.
- System should give insights on how many visitors are on particular course content and how many users have watched course content etc.