

EDUSHARE-RESOURCE SHARING APPLICATION

PROJECT SYNOPSIS

BATCHLOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY,

Mr. Tanishk Wagh (2165451242081)

Ms. Preeti Rajpure (2165451242053)

Mr. Omkar Wagh (2165451242072)

Mr. Mandar Shinde (2165451242069)

UNDER THE SUPERVISION OF

Mr . Pathak P.A



**ARVIND GAVALI COLLEGE OF ENGINEERING SATARA
2024-25**

TABLE OF CONTENTS

1. Introduction
2. Objectives and Scope of the Project
3. Literature Review
4. System Design (Methodology)
5. Hardware Required
6. Software Required
7. Estimation Of Project
8. Project Timeline
9. References

1. INTRODUCTION

Students often face problem while collecting notes and it interrupts their learning process. Our application will provide a platform for the students where they can come together for collaborative learning.

Students can share their notes/materials (both handwritten and in document - PPT, PDF, DOC, etc. format) to the admin of the website through the whatsapp or Email so the admin can check the notes and add them into the application and the students from any branch or semester from the different universities can access for free.

In our platform users can upload their notes giving a short description and some keywords related to the subject, branch and semester. Other users can search the notes using those keywords which will provide them filtered results. Each user will have an account through they will access the application.

The platform is planned to be a **LEARN | SHARE | EDUCATE** platform which aims to increase effectiveness and conceptual clarity of students.

2. OBJECTIVE AND SCOPE OF THE PROJECT

- **OBJECTIVE**

The objective of this proposed **Edushare-resource-sharing application** is to design and develop a user-friendly, AI-powered platform that enables students to easily share, discover, and access academic resources while fostering collaboration and knowledge exchange. The system aims to:

1. **Centralize Academic Resources:**
Provide a unified platform where students can upload, access, and organize various academic resources, including notes, project files, tutorials, and reference materials.
2. **Facilitate Collaboration:**
Promote interaction among students by incorporating features like resource rating, commenting, and discussion forums to encourage feedback and knowledge sharing.
3. **Ensure Seamless Accessibility:**
Support multiple file formats (e.g., PDFs, images, videos) and provide mobile-friendly access to enable resource sharing on-the-go.
4. **Provide Real-Time Support:**
Integrate an AI-driven chatbot to assist users in understanding platform features, navigating the website, and resolving common queries promptly.
5. **Maintain Data Security and Privacy:**
Implement secure data storage, user authentication, and controlled access mechanisms to ensure the integrity and confidentiality of uploaded resources.
6. **Support Scalability:**
Design the application to handle increasing user traffic and accommodate future expansion, including additional features like advanced analytics and integrations with educational tools.

- **SCOPE OF THE PROJECT**

User Registration and Authentication:

- Secure login and registration for students.
- Role-based access for students and administrators to manage features and data.

Resource Uploading and Sharing:

- Students can upload different types of educational resources (e.g., PDFs, Word documents, images).
- Resources are organized by categories, subjects, or types for easy navigation.

Resource Search and Download:

- A search feature to help students find resources based on keywords, categories, or resource types.
- Ability to view and download resources directly from the platform.

Database and Data Management:

- A secure database to store user profiles, uploaded resources, and platform activities.
- Admin panel for managing and reviewing content, user activities, and system health.

User Interface:

- A responsive and user-friendly interface designed using HTML, CSS, and JavaScript.
- Mobile-friendly design for access from various devices.

Security:

- Measures to ensure the confidentiality and integrity of users' data and shared resources.
- Secure authentication protocols to protect user accounts and resources.

Future Enhancements:

- Potential integration of more advanced AI features like auto-categorization of resources and predictive search.
- Addition of community collaboration tools like discussion forums or group sharing.

3. LITERATURE REVIEW

Sr.No	Author Name	Paper Name	Production Year	Technology Used
1	IEEE contributors	Empirical Study of Most Popular PHP Frameworks	2019	PHP frameworks (e.g., Laravel, CodeIgniter, Symfony) Proposes a PHP framework for improving modularity and performance
2	Xiaoming Chen, Jian Zhang	The Applications of PHP, HTML, and MySQL in Development of Website – Query Function	2021	PHP, MySQL, HTML Discusses PHP's integration with databases and web interfaces
3	<u>Helene L Teitelbaum</u>	Resource Sharing Platforms in the Digital Era	2023	Strategies for developing robust resource-sharing platforms using open-source PHP
4	IEEE contributors	A Systematic Review of Collaborative Digital Platforms	2023	PHP and Angular for designing platforms that enhance digital collaboration and resource sharing

5	Faculty of Language Studies and Human Development	An Urban and Rural Educational Resource Sharing and Exchange Platform Based on Cloud Platform Access Technology	2023	Cloud computing and network platforms Proposes a cloud-based educational resource-sharing platform designed to connect urban and rural schools, facilitating the exchange of high-quality educational materials
6	Majid Zamiri and Ali Esmaeili	Methods and Technologies for Supporting Knowledge Sharing within Learning Communities: A Systematic Literature Review	2023	Diverse methods including collaborative learning platforms and online forums
7	<u>Andreas F. Molisch</u> <u>Geoffrey Hinton</u>	Evolution of Resource Sharing Cooperation Based on Reciprocity in Peer-to-Peer Networks	2024	Examines how reciprocity models impact resource sharing in P2P social networks, providing insights into optimizing voluntary contributions and allocation mechanisms

GAP FINDING

In the **gap finding** process for the Resource Sharing Application, the focus is on identifying areas within the current educational resource-sharing landscape that are either underserved or not fully optimized. This analysis will help pinpoint the specific needs that the platform aims to address.

One of the key gaps is the **lack of a centralized platform** where students can easily share, access, and manage their educational resources. Many students still rely on informal methods of sharing, such as social media groups, email, or file-sharing services, which can be inefficient, disorganized, and difficult to navigate. The absence of a structured system means that students often struggle to find relevant materials or rely on outdated resources.

Another gap is the **limited personalization** in existing educational platforms. While many websites provide resources, they typically offer a one-size-fits-all approach, failing to tailor content based on individual needs, preferences, or academic progress. Students often have to manually search for resources, and the relevance of the search results is not always optimized, leading to time wasted in finding suitable materials. By integrating AI and machine learning, the application will provide personalized recommendations, enhancing resource discovery and ensuring that students have access to the most relevant content.

Additionally, there is often a **lack of interactivity and collaboration** in existing resource-sharing systems. Most platforms do not offer the ability for students to collaborate directly, ask questions, or discuss the resources shared. This limits the opportunity for peer-to-peer learning and knowledge exchange. The gap here lies in providing a platform where students not only share resources but can also interact, exchange ideas, and provide feedback on the resources, fostering a collaborative learning environment.

Finally, **security and privacy concerns** are often overlooked in many educational resource-sharing platforms. With the increasing number of online scams and data breaches, students need a secure environment to share and access materials. Existing platforms may not provide the level of data protection required to ensure that user information remains private and secure. The gap here is the lack of secure and trusted systems for data protection, which the resource sharing application will address through encryption and secure login methods.

4. SYSTEM DESIGN (METHODOLOGY)

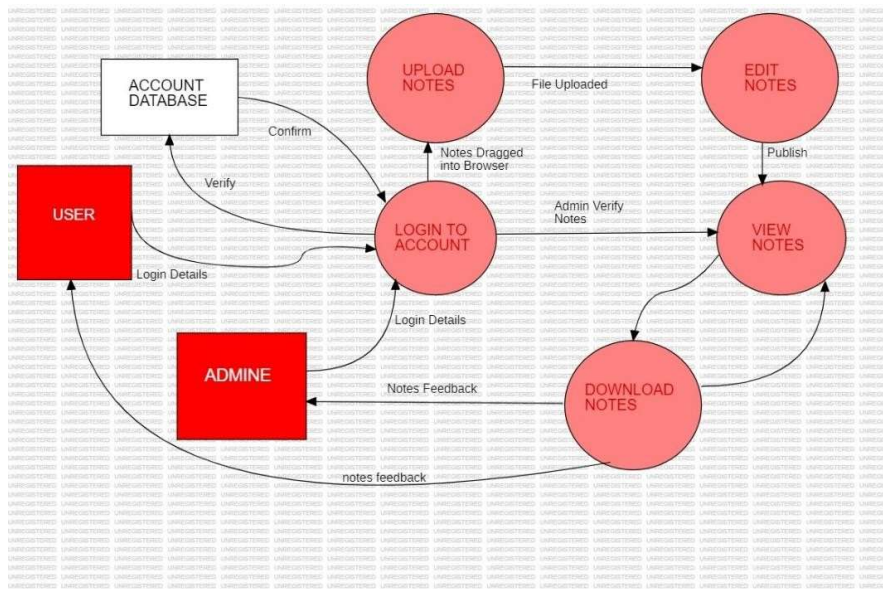


Fig :- Working Methodology Of Edushare -Resource Sharing Application

The methodology for developing the Resource Sharing Application follows a structured approach to ensure the creation of a robust, user-friendly, and efficient platform. It begins with requirements gathering, where in-depth research is conducted through surveys, interviews, and feedback from potential users. This phase focuses on understanding the needs, challenges, and expectations of students regarding resource sharing. By analyzing the current gaps in existing platforms, we can tailor the features to meet the specific demands of the user base.

Once the requirements are clear, the system design and architecture phase takes place. This includes developing a blueprint for the backend and frontend of the platform. A scalable and secure database is designed to store user information, uploaded resources, and activity logs. The platform's architecture is built with a focus on performance, scalability, and security, ensuring that the system can handle large volumes of users and resources efficiently. The frontend design emphasizes a clean, intuitive interface, making it easy for users to navigate and interact with the platform. This is achieved through wireframes and mockups, which are later translated into code using HTML, CSS, and JavaScript.

The next phase is frontend development, where the core user-facing features are built. These include secure login and registration systems, dashboards for uploading and managing resources, and a search feature that allows students to find materials based on categories, keywords, or file types. The search functionality integrates seamlessly with the backend, ensuring quick and accurate results. Additionally, a chatbot is integrated to guide users through the platform, helping them find resources or answer common questions in real time.

The backend development focuses on implementing the core logic and database interactions. This includes building the functionality for uploading, storing, and categorizing resources, as well as integrating machine learning and AI algorithms. The AI is used to provide personalized resource recommendations based on user activity, preferences, and trends, while also allowing the chatbot to respond intelligently to user queries. The backend also ensures that data is securely handled, with encryption protocols for protecting sensitive information.

5. HARDWARE REQUIRED

Development & Testing Environment

1. Developer Workstations

- Processor: Quad-core CPU (e.g., Intel i5 or AMD Ryzen 5)
- RAM: 16 GB
- Storage: 512 GB SSD
- Network: High-speed internet connection
- Operating System: Windows, macOS, or Linux

2. Local Development Servers (optional)

- Processor: Quad-core CPU
- RAM: 16 GB
- Storage: 512 GB SSD
- Network: High-speed internet connection

Production Environment

1. Web Server

- Processor: 8-core CPU (e.g., Intel Xeon or AMD EPYC)
- RAM: 32 GB or more
- Storage: 1 TB SSD
- Network: Gigabit Ethernet connection
- Operating System: Linux (Ubuntu, CentOS, etc.)

2. Database Server

- Processor: 8-core CPU
- RAM: 64 GB or more (dependent on the size of the database and number of concurrent connections)
- Storage: 2 TB SSD (consider NVMe for faster access)
- Network: Gigabit Ethernet connection
- Operating System: windows

3. File Storage Server (for storing uploaded notes)

- Processor: Quad-core CPU
- RAM: 16 GB
- Storage: Configurable based on storage needs (start with 2 TB HDD, scalable)

6. SOFTWARE REQUIRED

Functional Requirements

1. User Authentication and Authorization

- User registration (email, password, etc.)
- Login and logout functionality
- Role-based access control (admin, user, etc.)

2. User Profile Management

- View and edit personal profiles

3. Resource Management

- Create, edit, and delete resource listings
- Categorize resources (pdf , vedio links, mcq test links etc.)

4. Resource Search and Discovery

- Search resources by keywords, category, location, etc.
- Filter and sort search results
- View resource details

6. Notifications and Alerts

- notifications for important actions (request approval, new messages)
- In-app notifications for real-time updates

7. Messaging System

- Direct messaging between users

8. Admin Panel

- Manage users and resources
- View and manage system logs
- Handle disputes and issues

Non-Functional Requirements

1. Performance

- Fast load times (under 3 seconds for main pages)
- Efficient search and filtering capabilities

2. Scalability

- Support for a growing number of users and resources
- Ability to scale horizontally (adding more servers)

3. Security

- Data encryption (in transit and at rest)
- Protection against common vulnerabilities (SQL injection)
- Regular security audits

4. Usability

- Intuitive and user-friendly interface
- Mobile responsiveness

PROGRAMING LANGUAGE USED

1. Front-end Technologies

- HTML, CSS, JavaScript
- Frameworks: javascript

2. Back-end Technologies

- Programming language: php
- Frameworks: javascript

3. Database

- Relational Database: MySQL

4. APIs

- RESTful API for client-server communication
- Third-party APIs for notifications, payments (if applicable)

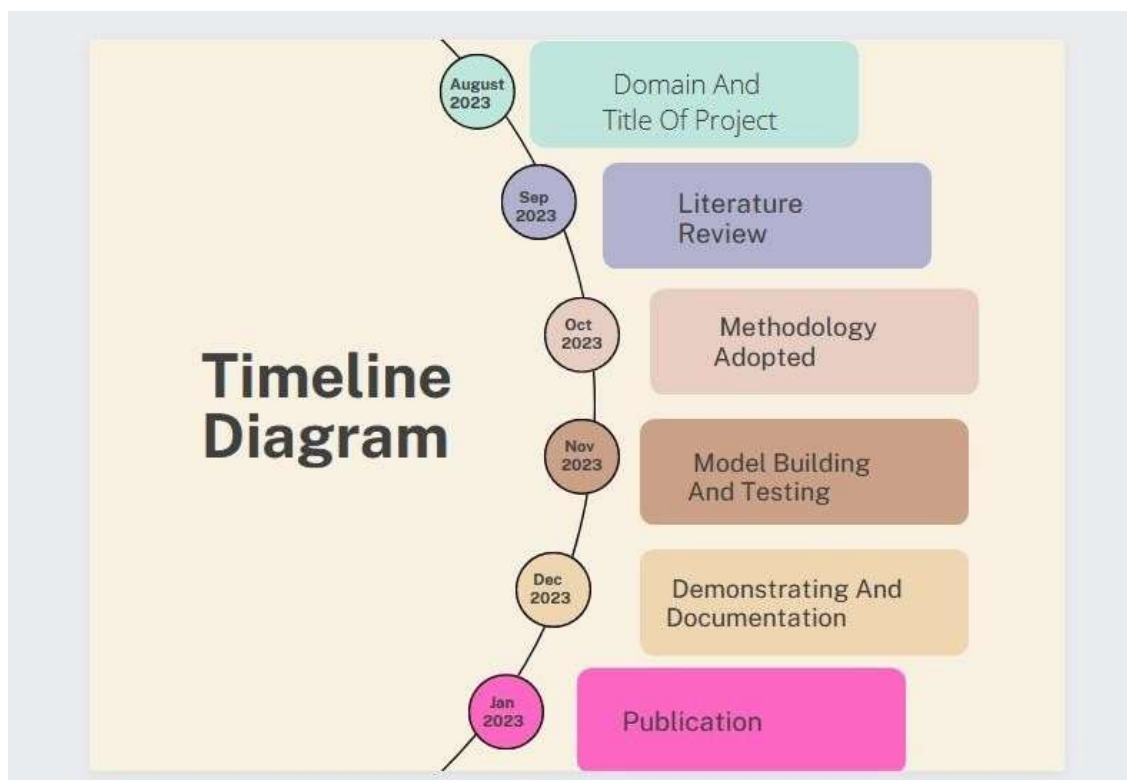
5. Hosting and Deployment

- Cloud platform: AWS, Google Cloud, or Azure
- CI/CD tools: Jenkins, GitHub Actions

7. ESTIMATION OF THE PROJECT

PROUDCT	UNIT	PRICE
XAMPP SERVER	1	-
VS-CODE	1	-
MYSQL DATABASE	1	-
STAR UML	1	-
TOTAL	4	10,000/-

8.TIMELINE



9. REFERENCES

1. IEEE contributors, “*Empirical Study of Most Popular PHP Frameworks*”, 2023. Discusses PHP frameworks (e.g., Laravel, CodeIgniter, Symfony) and proposes a PHP framework for improving modularity and performance.
2. Xiaoming Chen, Jian Zhang, “*The Applications of PHP, HTML, and MySQL in Development of Website – Query Function*”, 2023. Explores PHP's integration with databases and web interfaces.
3. Helene L. Teitelbaum, “*Resource Sharing Platforms in the Digital Era*”, 2023. Provides strategies for developing robust resource-sharing platforms using open-source PHP.
4. IEEE contributors, “*A Systematic Review of Collaborative Digital Platforms*”, 2023. Explores PHP and Angular for designing platforms that enhance digital collaboration and resource sharing.
5. Faculty of Language Studies and Human Development, “*An Urban and Rural Educational Resource Sharing and Exchange Platform Based on Cloud Platform Access Technology*”, 2021. Proposes a cloud-based educational resource-sharing platform to connect urban and rural schools.
6. Majid Zamiri, Ali Esmaeili, “*Methods and Technologies for Supporting Knowledge Sharing within Learning Communities: A Systematic Literature Review*”, 2024. Reviews diverse methods, including collaborative learning platforms and online forums.
7. Andreas F. Molisch, Geoffrey Hinton, “*Evolution of Resource Sharing Cooperation Based on Reciprocity in Peer-to-Peer Networks*”, 2019. Examines how reciprocity models impact resource sharing in P2P networks, optimizing voluntary contributions and allocation mechanisms.