Scholar connect: A platform to connect scholars using full stack

A PROJECT REPORT

Submitted by

Drawesh Yadav [RA2111003010757]
Shlok Goel [RA2111003010807]

Under the Guidance of

(Dr. M. Jagadeesh)

(Assistant Professor, Applications of AI| ML| DL, Software Engineering)

in partial fulfillment of the requirementsfor the degree of

BACHELOR OF TECHNOLOGIES

in

COMPUTER SCIENCE ENGINEERING

DEPARTMENT OF COMPUTATING TECHNOLOGIES

COLLEGE OF ENGINEERING AND TECHNOLOGIES

SRM INSTITUTE OF SCIENCE AND

TECHNOLOGIES KATTANKULATHUR- 603 203

NOVEMBER 2024

Department of Computing technologies

SRM Institute of Science & Technologies

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **Bachelor of Technologies in Computer Science Engineering**

Student Name : **Drawesh Yadav,Shlok Goel**

Registration Number : **RA2111003010757,RA21110030101807**

Title of Work : **Scholar Connect: A Platform To Connect Scholars using full stack**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook /University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGIES
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSP107L - Minor Project report titled "**Scholar Connect: A Platform To Connect Scholars using full stack**" is the bonafide work of "**Drawesh Yadav[RA2111003010757], Shlok Goel[RA2111003010807]"** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	SIGNATURE	SIGNATURE
DR. M. JAGAEESH	**Dr.R.S. Ponmagal**	**Dr. G. Niranjana**
SUPERVISOR ASSISTANT PROFESSOR DEPARTMENT OF COMPUTING TECHNOLOGIES	PANEL HEAD ASSOCIATE PROFESSOR DEPARTMENT OF COMPUTING TECHNOLOGIES	PROFESSOR &HEAD DEPARTMENT OF COMPUTING TECHNOLOGIES

SIGNATURE
Dr.K.Kishore Anthuvan Sahayaraj
PANEL MEMBER
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTING TECHNOLOGIES

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technologies, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. T. V. Gopal ,** Dean-CET, SRM Institute of Science and Technologies, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technologies, for her support throughout the project work.

We encompass our sincere thanks to , **Dr. M. Pushpalatha**, Professor and Associate Chairperson, School of Computing and **Dr. C.Lakshmi,** Professor and Associate Chairperson, School of Computing, SRM Institute of Science and Technologies, for their invaluable support.

We are incredibly grateful to our Head of the Department, **Dr. G. Niranjana**, Professor, Department of Computing technologies, SRM Institute of Science and Technologies, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Panel Head Dr.R.S.Ponmagal and Panel Members, Dr.K.Kishore Anthuvan Sahayaraj and Dr. M. Jagadeesh, Department of Computing technologies, SRM Institute of Science and Technologies, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, Saranya, Department of Computing technologies, SRM Institute of Science and Technologies, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, Mrs. Saranya S S and Dr. G. Manoj Kumar, Department of Computing technologies, SRM Institute of Science and Technologies, for providing us with an opportunity to pursue our project under his / her mentorship. He / She provided us with the freedom and support to explore the research topics of our interest. His / Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computing technologies, S.R.M Institute of Science and Technologies, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement

Authors

ABSTRACT

In today's fast-evolving digital world, there is an increasing demand for a unified knowledge-sharing platform that effectively merges collaborative software development with academic research dissemination. This paper presents the design and implementation of a hybrid platform that combines the core functionalities of GitHub, a widely-used environment for software development and version control, with the scholarly communication features found in academic repositories such as Google Scholar. The platform is designed to streamline workflows for researchers and developers alike, offering a space where both code and academic contributions can coexist and evolve in a collaborative ecosystem. By integrating these traditionally separate domains, the platform encourages interdisciplinary collaboration, allowing users to manage code development, maintain version history, and share academic publications within a single environment. This fusion enables seamless contributions to both academic literature and software projects, enhancing productivity and promoting innovation across scientific and engineering communities.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction TO Scholar Connect:

The increasing digitization of academic research and the integration of software development in modern scientific practices demand platforms that streamline collaboration, resource management, and communication. Traditional academic repositories such as Google Scholar and institutional archives focus on disseminating scholarly publications, while platforms like GitHub have revolutionized software development through features like version control and collaborative coding. However, there remains a significant gap between the scholarly communication systems used by researchers and the technical infrastructure required for collaborative software development. The complexity of managing both research output and code in separate environments can hinder interdisciplinary work, especially in fields where research and software development intersect, such as computational science, bioinformatics, and data-driven engineering. To address this gap, we present Scholar Connect, a unified platform designed to facilitate collaboration between researchers and developers, enhancing both academic productivity and software development workflows. The platform integrates the core functionalities of academic repositories, such as user profile management and research paper dissemination, with features inspired by version control platforms like GitHub, enabling users to collaboratively manage both code and research output in a seamless environment. Scholar Connect's goal is to foster interdisciplinary collaboration by providing a space where academic and technical contributions coexist, promoting innovation and accelerating the pace of discovery in various research fields.

At the core of Scholar Connect is its personalized dashboard, which allows users to maintain professional profiles, upload academic papers, and share resources with their peers. A key feature of the platform is the ability to manage software development projects, including version history and collaborative coding, directly alongside academic publications. This enables researchers to not only publish their work but also make their associated codebases available for use, review, and further development by the scholarly community. The ability to integrate academic research with software development enhances transparency and reproducibility, which are essential pillars of modern scientific practice.

1.2 Motivation

The motivation behind developing this integrated knowledge-sharing platform stems from the need to bridge the gap between software development and academic research, which are often handled in separate ecosystems. In today's digital world, both academic researchers and developers contribute significantly to knowledge creation, yet their workflows are fragmented. While developers use tools like GitHub for version control and collaboration on software projects, researchers rely on academic repositories such as Google Scholar to publish, share, and access research papers.

Key Motivations:

1. **Streamlined Workflow for Researchers and Developers: Combining code development and scholarly communication into a single platform eliminates the need for users to switch between multiple tools. Researchers and developers can now collaborate on code, share research results, and link their work directly to academic publications, enabling smoother project management.**
2. **Fostering Interdisciplinary Collaboration: The platform promotes collaboration across different disciplines, encouraging researchers and developers to work together more efficiently. This is especially useful for scientific and engineering communities, where code and research are often interdependent, such as in data science, artificial intelligence, and computational biology.**
3. **Improved Research Accessibility and Transparency: By merging academic research with open-source software development, the platform fosters open science practices. Researchers can share code along with their publications, making their methodologies more transparent and reproducible. This enhances the overall integrity of research and accelerates the scientific discovery process.**

This project is motivated by the desire to create an integrated, efficient, and collaborative ecosystem that empowers users to contribute to both academic research and software development, fostering innovation and interdisciplinary collaboration.

1.3 Sustainable Development Goal of the Project

The Scholar Connect platform aligns with several **Sustainable Development Goals (SDGs)** outlined by the United Nations. These goals aim to promote sustainable, inclusive development across various sectors, including education, innovation, and collaboration. Here's how this project contributes to specific SDGs:

1. Goal 4: Quality Education

- **Promoting Accessible Knowledge and Learning**: By integrating academic research and code development into a single platform, this project encourages the free sharing of knowledge, which supports inclusive and equitable quality education. It allows educators, students, and researchers to access up-to-date academic publications and open-source software projects, fostering continuous learning and development.
- **Open Access to Research**: The platform enables researchers to share their work more broadly, promoting open access to educational resources that can benefit learners worldwide, especially in underprivileged areas.

Chapter 2

LITERATURE SURVEY

2.1 Limitations Identified from Literature Survey

1. Unified Platform for Research, Code, and Collaboration Tools
    - Existing studies on code generation tools (like Ziegler et al., 2022; Nguyen & Nadi, 2022) and GitHub Copilot's code performance have primarily focused on software development tasks or educational applications. They do not address a holistic platform that combines academic papers, code sharing, and collaborative features in a single ecosystem. This represents a gap where Scholar Connect could unify code and paper collaboration with tailored tools for interdisciplinary academic use.
2. Cross-Disciplinary Code and Research Sharing
    - While CoProtector by Sun et al. (2022) examines code protection for open-source software, this paper and others do not tackle the broader challenge of managing and securing both code and data across interdisciplinary fields. In Scholar Connect, code from various domains (e.g., social sciences, bioinformatics) can coexist with research papers. This missing aspect of cross-disciplinary integration—especially in terms of security and accessibility—is a significant gap that this platform could fill.

LITERATURE SURVEY

2.2 Research Objective

1. Develop an Integrated Platform for Code and Research Collaboration
    - Create a unified environment where users can collaboratively work on research publications, code, and datasets. This will allow interdisciplinary research teams to easily access and share resources, filling the current gap between traditional research platforms and code repositories.
2. Enhance Citation Tracking Across Multiple Research Formats
    - Design a citation and attribution system that supports not only traditional papers but also code, datasets, and other interdisciplinary research outputs. This objective addresses the need for a comprehensive system that credits all forms of academic contributions, encouraging proper recognition across varied formats.
3. Improve Real-Time, Multi-Format Collaborative Features
    - Implement collaborative tools that allow synchronous editing across formats, including research papers, code, and data files. This supports interdisciplinary teams by providing a more integrated experience, enabling users to work together in real-time on all aspects of a research project.

LITRATURE SURVEY

2.3 Project Backlog

Table 2.1

S.No	User Stories of Scholar connect
#US1	As a new user, I want to easily register for the platform so that I can gain access to its features like Profile creation.
#US 2	As a new user, I want to access my profile information and see my uploaded material
#US 3	As a new user, I want to be able to be able to upload my research paper
#US 4	As a user, I want to be able to chat with different peoples of different disciplines.
#US 5	As a user, I want to know other's opinions on my research paper
#US 6	As a user, I want to participate in peer-to-peer learning sessions so that I can gain knowledge from others and share my expertise in a collaborative environment.
#US 7	As a user, I want to provide feedback on the platform's features and content so that I can help improve the user experience and ensure the content meets my needs.

LITRATURE SURVEY

2.4 Plan of Action

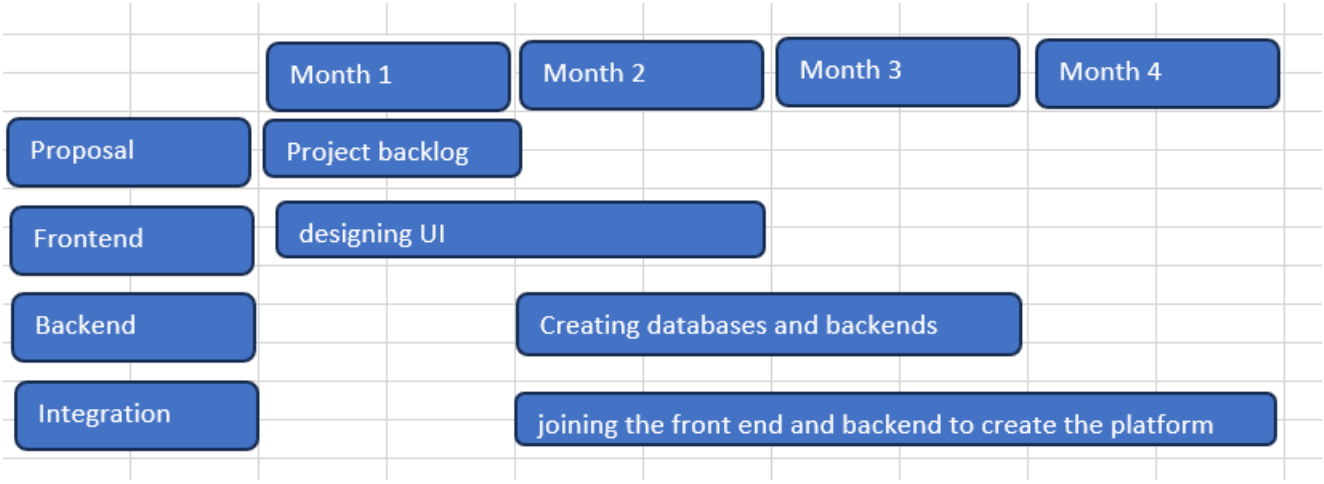


Figure 2.1

CHAPTER 3

SPRINT PLANNING AND EXECTION METHODOLOGY

3.1 Sprint 1

3.1.1 Objectives with user stories of Sprint I

Objective 1: Define User Stories and Build the Project Backlog

- User Story 1: As a new user, I want to create an account so I can access the platform's features and collaborate on research projects.
- User Story 2: As a researcher, I want to log in securely so I can resume work on my projects and maintain privacy of my data.
- User Story 3: As a user, I want to upload and view my research documents and code in one place, so I don't need multiple platforms.
- Goal: Develop clear, prioritized user stories to ensure the project backlog covers essential features and aligns with user needs.

Objective 2: Implement the Frontend Framework and Core Navigation

- User Story 7: As a developer, I want the frontend set up with reusable components, so adding features in future sprints is streamlined.
- User Story 8: As a user, I want responsive navigation between sections like profile, projects, and dashboard to support my workflow.
- Goal: Set up the frontend using a framework (e.g., React or Vue) with reusable components and basic navigation between core sections.

Objective 3: Develop Basic Frontend Interactivity for User Actions

- User Story 9: As a user, I want interactive elements (like buttons and forms) to respond quickly, so my experience feels smooth and engaging.
- User Story 10: As a collaborator, I want search and filter options to find relevant projects and collaborators easily.
- Goal: Implement basic interactivity, including buttons, forms, and navigation bars to give users a feel of the platform's functionality.

3.1 Sprint 1

3.1.2 Functional Document

1. Introduction

In Sprint 1 of Scholar Connect, the objective is to establish the foundational elements of the platform by defining user stories, creating a project backlog, and developing the user interface (UI) and frontend structure. This sprint will ensure that all core features required for user interaction, navigation, and basic functionality are planned and designed to serve as a base for future development.

2. Product Goal

The goal of this platform is to streamline workflows for both software developers and researchers, allowing them to collaboratively develop code, manage version history, and share academic research within a single environment. This hybrid system will encourage interdisciplinary collaboration and reduce inefficiencies in knowledge sharing by combining version control tools and academic publication repositories.

3. Demography (Users Location)

- Target Users: Researchers, software developers, industry professionals, academic institutions.
- User Characteristics: Varied backgrounds including academic researchers, open-source contributors, and corporate developers. Users range from technical experts in software development to non-technical researchers.
- Target Location: Global usage with primary interest in academic and research communities, tech startups, and open-source projects.

4.Business Process

The business process for Sprint 1 of Scholar Connect focuses on user management, project organization, and enhancing discoverability. Upon registration, users set up profiles and can sign in via traditional registration or third-party OAuth options (e.g., GitHub, Google Scholar). Post-authentication, users access a personalized dashboard summarizing activities, projects, and recent collaborations.

For project management, users create repositories to upload research papers and code, set permissions, and track versions. Publications can be linked to projects, allowing users to view citations and the academic impact associated with their work. Through collaborative tools,

users invite contributors, assign role-based permissions, and use messaging to coordinate and leave comments. These tools enhance interdisciplinary collaboration within secure, managed access.

Search and discoverability allow users to locate relevant research or collaborators through keyword and filter-based searches, supporting interdisciplinary outreach. Finally, feedback collection through forms and usability tests enables iterative improvements to Scholar Connect, aligning with evolving user needs.

These processes ensure that Scholar Connect serves as an integrated platform for managing research, collaboration, and scholarly recognition, laying a strong foundation for future development phases.

5.Features

Here are some potential features for Scholar Connect platform that merges collaborative software development with academic research dissemination:

Core Features

1. User Profiles:

- Profiles for researchers and developers with customizable information, including their areas of expertise, publications, and projects.

2. Project Repositories:

- Allow users to create, manage, and share repositories for both code and related academic publications.

3. Interdisciplinary Collaboration:

- Tools to connect researchers from different fields for collaborative projects.
- Networking features to follow and connect with other users based on interests.

3. Resource Sharing:

- A section for sharing datasets, tools, and other resources relevant to specific projects or academic fields.

3.1 Sprint 1

3.1.3 Architecture Document

1. Architecture Overview

The Scholar Connect platform leverages a microservices architecture to enable independent development, deployment, and scaling of various components such as user management, project repositories, and publication management. Each microservice communicates via lightweight APIs, promoting flexibility and maintainability. The platform also implements an event-driven architecture to facilitate asynchronous communication and real-time updates, enhancing collaboration between developers and researchers.

2. System Overview

2.1 Purpose

To provide a unified platform that integrates collaborative software development with academic research dissemination.

2.2 Key Features

- User Profiles
- Project Repositories
- Integrated Publication Management
- Collaboration Tools

3. Architectural Styles

3.1 Microservices Architecture

- Description: The platform is designed as a collection of small, independently deployable services, each responsible for a specific business function (e.g., user authentication, project management). This architecture enables teams to develop, deploy, and scale services independently, promoting resilience and agility.
- Components:
    - User Service: Manages user profiles and authentication.
    - Project Service: Handles project repositories and version control.
    - Publication Service: Manages academic publications and metadata.

3.2 Event-Driven Architecture

- Description: The platform utilizes an event-driven approach to enable real-time updates and notifications. Services communicate by publishing and subscribing to events,

allowing for decoupled interactions and improved responsiveness.

- Components:
  - Event Bus Facilitates communication between services by passing messages/events.
  - Event Listeners: Services that react to specific events (e.g., a new publication uploaded triggers notifications to relevant users).

3.1 Sprint 1

3.1.4 Result Analysis

Sprint Goals

- Focus on defining user stories and project backlog.
- Develop the UI and front-end components for the initial features of Scholar Connect.

Front end work done according to the Timeline

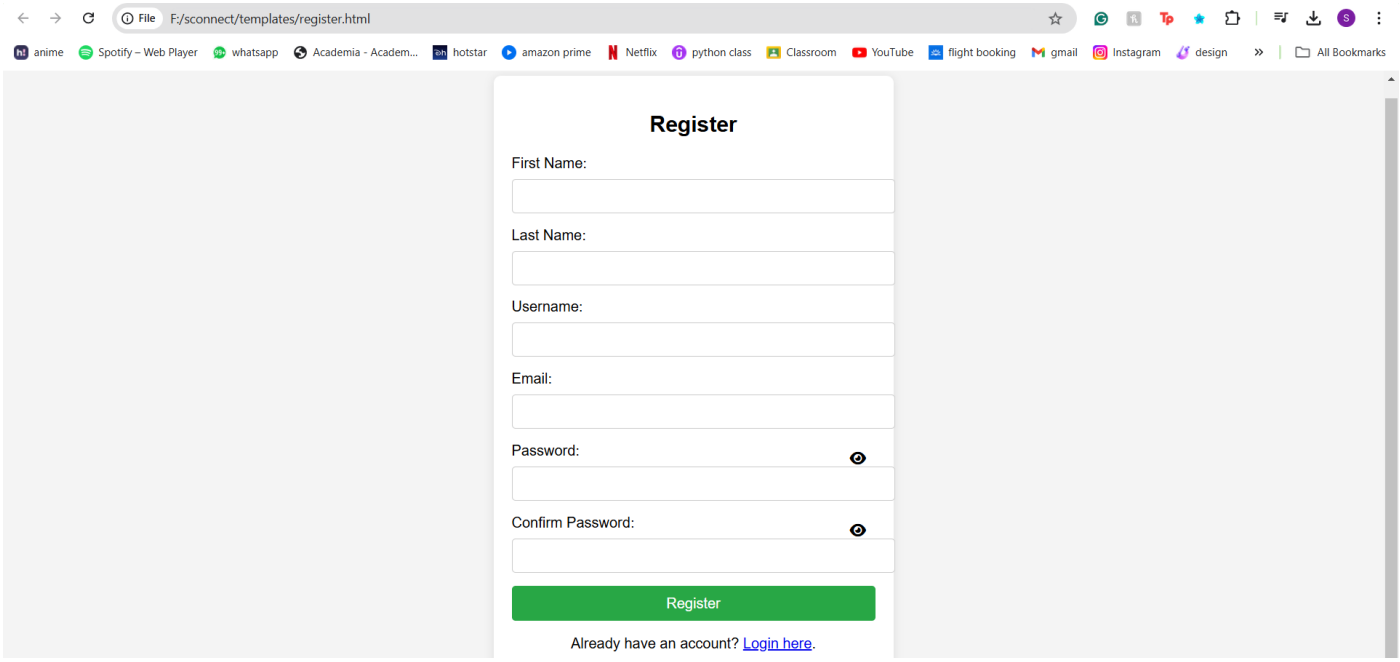


Figure 1

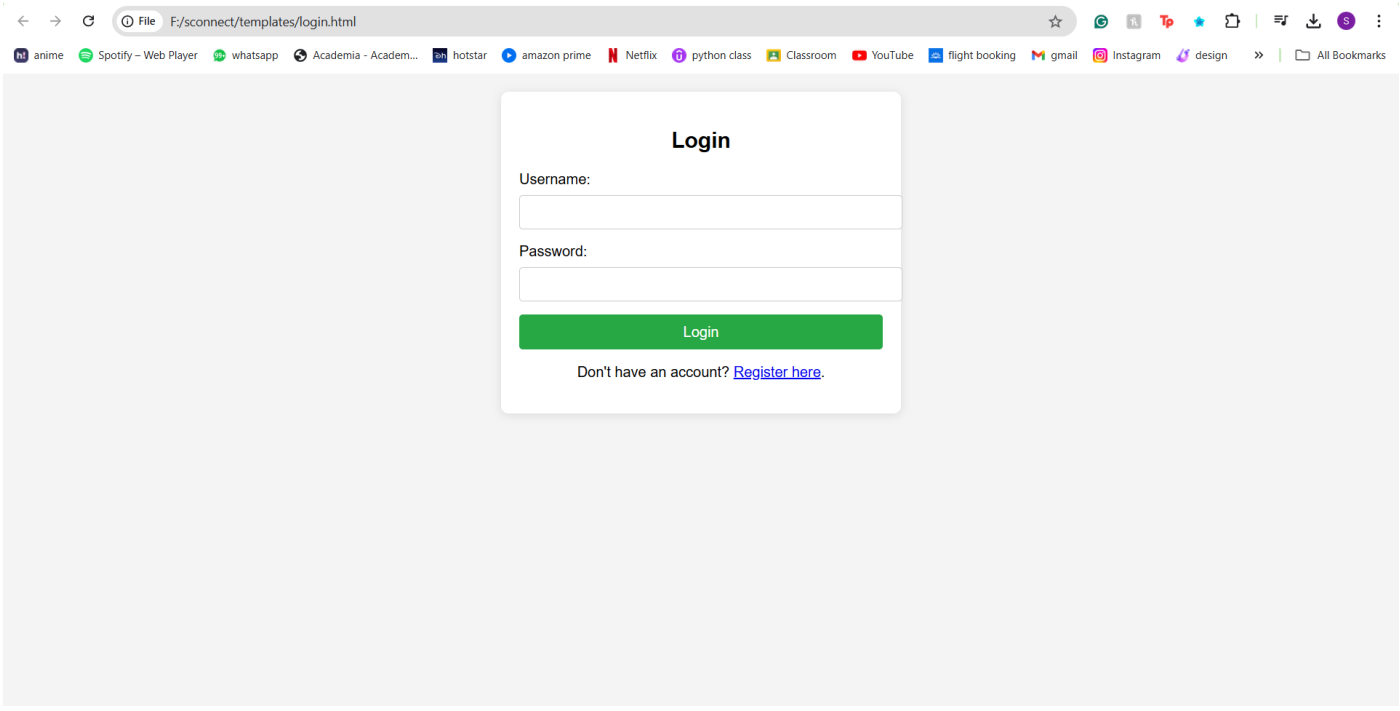


Figure 2

3.1 Sprint 1

3.1.5 Sprint Retrospective

Sprint Retrospective			
Liked	**Learned**	**Lacked**	**Longed For**
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
Team members appreciated the collaborative environment during the sprint, where everyone contributed ideas during discussions about user stories and UI design.	Early feedback from stakeholders highlighted the value of involving users in the design process, emphasizing that incorporating their insights leads to a more user-friendly product.	The team experienced challenges with time allocation, particularly in completing the project creation UI. Better time estimates could have improved overall task completion.	The team expressed a desire for better tools to facilitate collaboration and communication, such as a dedicated platform for sharing design mockups and project updates.
he process of defining and refining user stories was well-received, as it helped clarify the scope of work and align the team's efforts with user needs.	The experience of prioritizing and refining the project backlog taught the team the importance of regularly reassessing priorities to ensure alignment with user needs and project goals.	There was a lack of comprehensive testing, particularly for edge cases in the user registration and login processes. More time dedicated to testing could have identified issues earlier.	There was a longing for more frequent feedback sessions to iterate on the design and functionality before finalizing features.
The initial UI components, such as the registration and login forms, received positive feedback for their aesthetics and usability. The team enjoyed creating a visually appealing interface.	The team learned about potential integration issues between the front end and back end, emphasizing the need for thorough testing and communication between teams early in the development process.	The UI was not fully optimized for mobile devices, highlighting a gap in considering responsive design from the outset.	Team members looked forward to tackling more advanced features in the next sprint, such as collaboration tools and publication management, to enhance the platform's capabilities.

Table 1

3.2 Sprint 2

3.2.1 Objectives with user stories of Sprint II

Objective 1: Design and Implement Database Schemas

- User Story 1: As a developer, I want to create a user table in the database so that I can store user profiles and their information securely.
- User Story 2: As a developer, I want to create a project table linked to user profiles so that I can manage project data associated with each user.
- User Story 3: As a developer, I want to create a publication table that connects to projects so that users can manage their research documents in one place.
- Goal: Establish a well-structured database schema that supports user profiles, project repositories, and publication management.

Objective 2: Ensure Data Integrity and Security

- User Story 1: As a developer, I want to implement foreign key constraints between user, project, and publication tables so that data integrity is maintained.
- User Story 2: As an administrator, I want to define user roles and permissions to control access to sensitive data, ensuring that only authorized users can view or edit information.
- Goal: Implement robust data validation and security measures to protect user information and ensure consistent data relationships.

Objective 3: Facilitate Data Initialization and Migration

- User Story 1: As a project administrator, I want to set up initial data in the database so that I have a baseline for testing and development purposes.
- User Story 2: As a developer, I want to ensure that any existing data is migrated to the new database structure without loss or corruption so that users can continue working seamlessly.
- Goal: Ensure a smooth transition to the new database by populating it with initial data and migrating any necessary existing data.

3.2 Sprint 2

3.2.2 Functional Document

1.Indroduction

In Sprint 2 of Scholar Connect, the objective is to build upon the foundational elements established in Sprint 1 by designing and implementing the database architecture that will support user profiles, project repositories, and publication management. This sprint focuses on creating a robust database schema, ensuring data integrity and security, and developing API endpoints for seamless interaction between the front end and back end. By establishing a structured database and facilitating efficient data management, this sprint will lay the groundwork for a more integrated and functional platform, enhancing user experience and paving the way for advanced features in future sprints.

2. Product Goal

The goal of this platform is to streamline workflows for both software developers and researchers, allowing them to collaboratively develop code, manage version history, and share academic research within a single environment. This hybrid system will encourage interdisciplinary collaboration and reduce inefficiencies in knowledge sharing by combining version control tools and academic publication repositories.

3. Demography (Users Location)

- Target Users: Researchers, software developers, industry professionals, academic institutions.
- User Characteristics: Varied backgrounds including academic researchers, open-source contributors, and corporate developers. Users range from technical experts in software development to non-technical researchers.
- Target Location: Global usage with primary interest in academic and research communities, tech startups, and open-source projects.

4.Business Process

The business process for Sprint 1 of Scholar Connect focuses on user management, project organization, and enhancing discoverability. Upon registration, users set up profiles and can sign in via traditional registration or third-party OAuth options (e.g., GitHub, Google Scholar). Post-authentication, users access a personalized dashboard summarizing activities, projects, and recent collaborations.

For project management, users create repositories to upload research papers and code, set permissions, and track versions. Publications can be linked to projects, allowing users to view citations and the academic impact associated with their work. Through collaborative tools, users invite contributors, assign role-based permissions, and use messaging to coordinate and leave comments. These tools enhance interdisciplinary collaboration within secure, managed access.

Search and discoverability allow users to locate relevant research or collaborators through keyword and filter-based searches, supporting interdisciplinary outreach. Finally, feedback collection through forms and usability tests enables iterative improvements to Scholar Connect, aligning with evolving user needs.

These processes ensure that Scholar Connect serves as an integrated platform for managing research, collaboration, and scholarly recognition, laying a strong foundation for future development phases.

5.Features

Here are some potential features for Scholar Connect platform that merges collaborative software development with academic research dissemination:

Core Features

4. User Profiles:
- Profiles for researchers and developers with customizable information, including their areas of expertise, publications, and projects.

5. Project Repositories:
- Allow users to create, manage, and share repositories for both code and related academic publications.

3. Interdisciplinary Collaboration:
- Tools to connect researchers from different fields for collaborative projects.
- Networking features to follow and connect with other users based on interests.

6. Resource Sharing:
- A section for sharing datasets, tools, and other resources relevant to specific projects or academic fields.

3.2 Sprint 2

3.2.3 Architecture Document

1. Architecture Overview

The Scholar Connect platform leverages a microservices architecture to enable independent development, deployment, and scaling of various components such as user management, project repositories, and publication management. Each microservice communicates via lightweight APIs, promoting flexibility and maintainability. The platform also implements an event-driven architecture to facilitate asynchronous communication and real-time updates, enhancing collaboration between developers and researchers.

2. System Overview

2.1 Purpose

To provide a unified platform that integrates collaborative software development with academic research dissemination.

2.2 Key Features

- User Profiles
- Project Repositories
- Integrated Publication Management
- Collaboration Tools

3. Architectural Styles

3.1 Microservices Architecture

- Description: The platform is designed as a collection of small, independently deployable services, each responsible for a specific business function (e.g., user authentication, project management). This architecture enables teams to develop, deploy, and scale services independently, promoting resilience and agility.
- Components:
    - User Service: Manages user profiles and authentication.
    - Project Service: Handles project repositories and version control.
    - Publication Service: Manages academic publications and metadata.

3.2 Event-Driven Architecture

- Description: The platform utilizes an event-driven approach to enable real-time updates and notifications. Services communicate by publishing and subscribing to events,

allowing for decoupled interactions and improved responsiveness.

- Components:
  - Event Bus Facilitates communication between services by passing messages/events.
  - Event Listeners: Services that react to specific events (e.g., a new publication uploaded triggers notifications to relevant users).

3.2 Sprint 2

3.2.4 Result Analysis

Sprint Goals

- Design and implement a robust database schema.
- Ensure data integrity and security.
- Develop API endpoints for database interaction.
- Facilitate data initialization and migration.

Database and model development work done as per objectives and timeline.

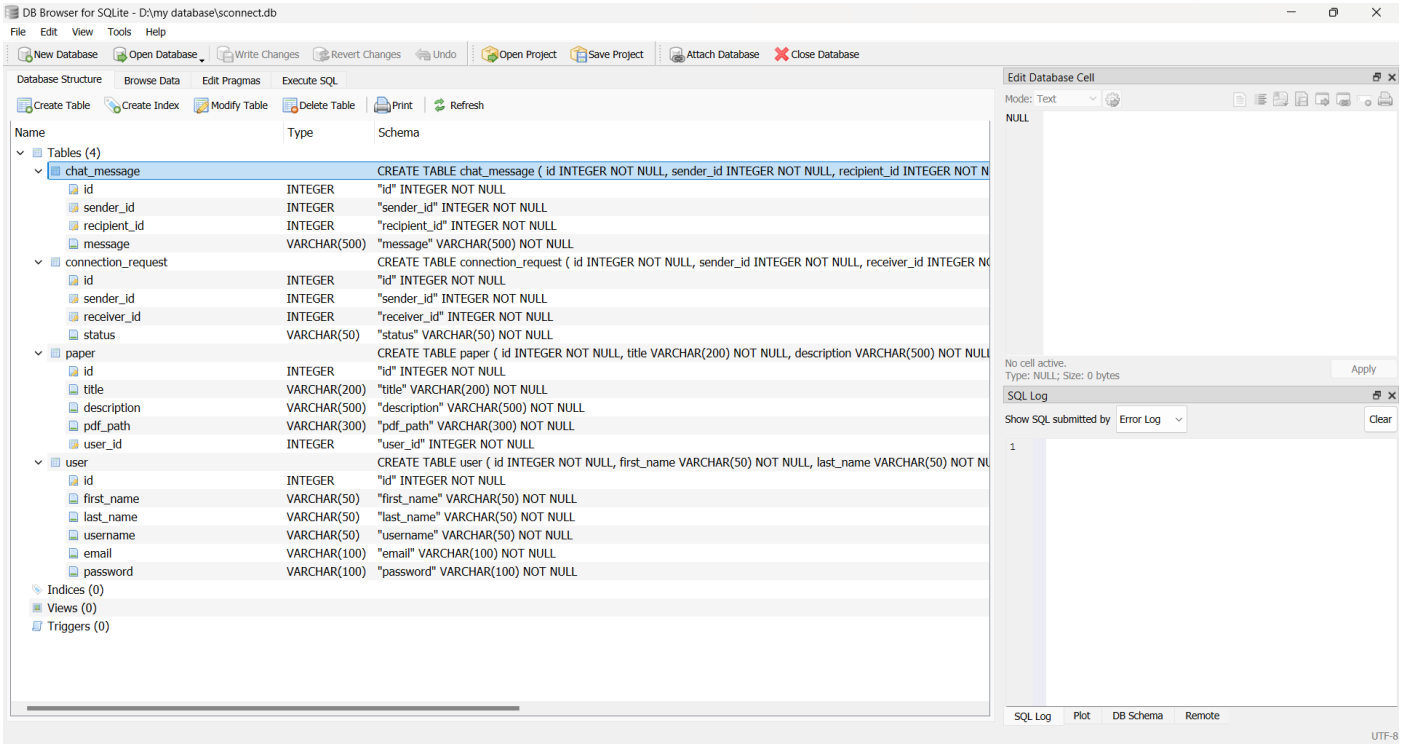


Figure 3

3.2 Sprint 2

3.2.5 Sprint Retrospective

Table 2

	Sprint Retrospective		
Liked	**Learned**	**Lacked**	**Longed For**
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
Team members appreciated the clarity that a well-defined database schema provided for the application.	The team learned about the importance of database normalization to reduce redundancy and maintain data integrity.	The initial documentation of the database schema and relationships was insufficient, making it challenging for new team members to understand the structure quickly.	The team expressed a desire for additional time to conduct thorough testing of database functionalities and queries.
The process of designing and implementing a database allowed team members to enhance their SQL and database management skills.	ained insights into writing efficient SQL queries for data retrieval and management.	There was a lack of automated testing procedures for validating database integrity and queries, which could have improved reliability.	Longed for structured training sessions on advanced database management techniques and optimization strategies.
Positive feedback on how the team effectively collaborated to solve database design challenges.	Learned the value of integrating feedback from stakeholders early in the database design process to ensure alignment with project requirements.	Insufficient focus on performance testing of queries and database interactions, leading to potential bottlenecks.	eam members wished for better tools to visualize the database schema and relationships, aiding understanding and communication.

3.3 Sprint 3

3.3.1 Objectives with user stories of Sprint III

Objective 1: Develop Backend Services

- User Story 1: As a developer, I want to create the server-side logic for user authentication so that users can securely log in and access their accounts.
- User Story 2: As a developer, I want to implement business logic for project management so that users can create, update, and delete their projects efficiently.
- User Story 3: As a developer, I want to develop publication management functionalities so that users can upload and manage their research documents seamlessly.
- Goal: Build robust backend services that handle user interactions and data management effectively.

Objective 2: Integrate Frontend with Backend APIs

- User Story 1: As a front-end developer, I want to connect the user registration form to the backend API so that new users can create accounts through the UI.
- User Story 2: As a front-end developer, I want to integrate project management features with backend services so that users can view, add, and modify their projects in real-time.
- User Story 3: As a front-end developer, I want to link publication uploads to the backend so that users can seamlessly manage their research documents from the UI.
- Goal: Ensure seamless communication between the front end and back end, allowing for dynamic user interactions.

Objective 3: Conduct Testing and Quality Assurance

- User Story 1: As a QA tester, I want to perform integration testing on all API endpoints to ensure they work correctly with the front end.
- User Story 2: As a QA tester, I want to identify and report any bugs or performance issues during testing so that the team can address them before the next sprint.
- Goal: Ensure a stable and functional integration of all components through thorough testing and quality assurance processes.

3.3 Sprint 3

3.3.1 Functional Document

1. Introduction

In Sprint 3 of Scholar Connect, the objective is to build upon the foundational elements established in the previous sprints by developing the backend services and integrating them with the frontend. This sprint focuses on creating the server-side logic that will handle user authentication, project management, and publication management. By implementing robust APIs and ensuring seamless communication between the frontend and backend, this sprint aims to enhance the platform's functionality and user experience. Additionally, thorough testing will be conducted to identify and resolve any issues, ensuring a stable and reliable system as we move forward in the development process.

2. Product Goal

The goal of this platform is to streamline workflows for both software developers and researchers, allowing them to collaboratively develop code, manage version history, and share academic research within a single environment. This hybrid system will encourage interdisciplinary collaboration and reduce inefficiencies in knowledge sharing by combining version control tools and academic publication repositories.

3. Demography (Users Location)

- Target Users: Researchers, software developers, industry professionals, academic institutions.
- User Characteristics: Varied backgrounds including academic researchers, open-source contributors, and corporate developers. Users range from technical experts in software development to non-technical researchers.
- Target Location: Global usage with primary interest in academic and research communities, tech startups, and open-source projects.

4.Business Process

The business process for Sprint 1 of Scholar Connect focuses on user management, project organization, and enhancing discoverability. Upon registration, users set up profiles and can sign in via traditional registration or third-party OAuth options (e.g., GitHub, Google Scholar). Post-authentication, users access a personalized dashboard summarizing activities, projects, and recent collaborations.

For project management, users create repositories to upload research papers and code, set permissions, and track versions. Publications can be linked to projects, allowing users to view citations and the academic impact associated with their work. Through collaborative tools, users invite contributors, assign role-based permissions, and use messaging to coordinate and leave comments. These tools enhance interdisciplinary collaboration within secure, managed access.

Search and discoverability allow users to locate relevant research or collaborators through keyword and filter-based searches, supporting interdisciplinary outreach. Finally, feedback collection through forms and usability tests enables iterative improvements to Scholar Connect, aligning with evolving user needs.

These processes ensure that Scholar Connect serves as an integrated platform for managing research, collaboration, and scholarly recognition, laying a strong foundation for future development phases.

5.Features

Here are some potential features for Scholar Connect platform that merges collaborative software development with academic research dissemination:

Core Features

7. User Profiles:

- Profiles for researchers and developers with customizable information, including their areas of expertise, publications, and projects.

8. Project Repositories:

- Allow users to create, manage, and share repositories for both code and related academic publications.

3. Interdisciplinary Collaboration:

- Tools to connect researchers from different fields for collaborative projects.
- Networking features to follow and connect with other users based on interests.

9. Resource Sharing:

- A section for sharing datasets, tools, and other resources relevant to specific projects or academic fields.

3.3 Sprint 3

3.3.3 Architecture Document

1. Architecture Overview

The Scholar Connect platform leverages a microservices architecture to enable independent development, deployment, and scaling of various components such as user management, project repositories, and publication management. Each microservice communicates via lightweight APIs, promoting flexibility and maintainability. The platform also implements an event-driven architecture to facilitate asynchronous communication and real-time updates, enhancing collaboration between developers and researchers.

2. System Overview

2.1 Purpose

To provide a unified platform that integrates collaborative software development with academic research dissemination.

2.2 Key Features

- User Profiles
- Project Repositories
- Integrated Publication Management
- Collaboration Tools

3. Architectural Styles

3.1 Microservices Architecture

- Description: The platform is designed as a collection of small, independently deployable services, each responsible for a specific business function (e.g., user authentication, project management). This architecture enables teams to develop, deploy, and scale services independently, promoting resilience and agility.
- Components:
    - User Service: Manages user profiles and authentication.
    - Project Service: Handles project repositories and version control.
    - Publication Service: Manages academic publications and metadata.

3.2 Event-Driven Architecture

- Description: The platform utilizes an event-driven approach to enable real-time updates and notifications. Services communicate by publishing and subscribing to events,

allowing for decoupled interactions and improved responsiveness.

- Components:
  - Event Bus Facilitates communication between services by passing messages/events.
  - Event Listeners: Services that react to specific events (e.g., a new publication uploaded triggers notifications to relevant users).

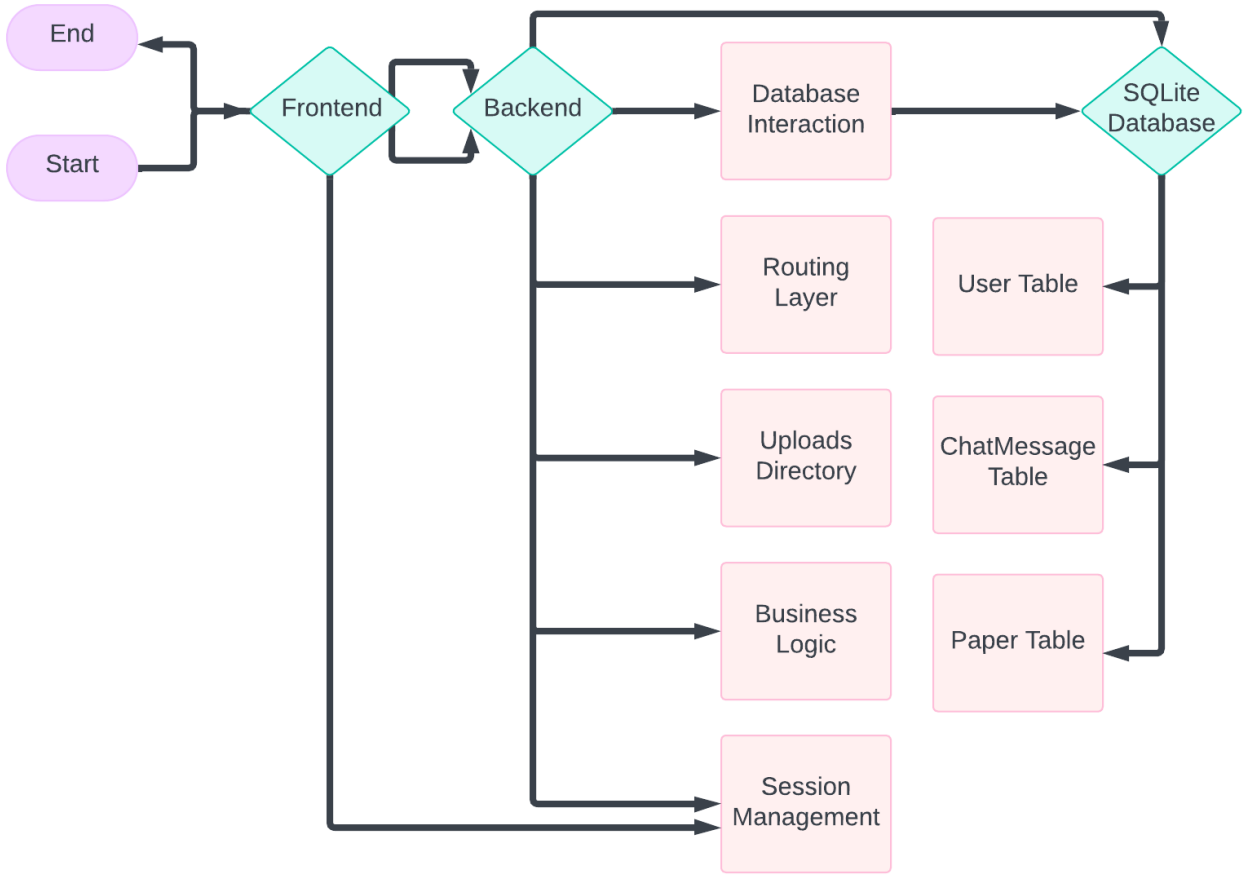


Figure 4

3.3 Sprint 3

3.3.4 Result Analysis

Sprint Goals

- Develop the backend services necessary for the Scholar Connect platform, including server-side logic and database interactions.
- Integrate the front end with the backend services, ensuring seamless communication between the two.
- Implement authentication and authorization mechanisms to secure user data and interactions.
- Test the integrated components to ensure functionality and performance.

Backend and model development work done as per objectives and timeline.

```python
from flask import Flask, render_template, request, redirect, url_for, flash, session
from flask_sqlalchemy import SQLAlchemy
from werkzeug.utils import secure_filename
import os
from flask_session import Session
from flask_login import LoginManager, UserMixin, login_required, current_user, login_user, logout_user
from flask_bcrypt import Bcrypt

app = Flask(__name__)
app.secret_key = 'b5a1f4a2c79dbf57b20f02d0e01f5c92'

# Initialize Bcrypt
bcrypt = Bcrypt(app)

# SQLite Database configuration
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///D:/my database/sconnect.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

# Server-side session configuration
app.config['SESSION_TYPE'] = 'filesystem'
app.config['SESSION_PERMANENT'] = False
app.config['SESSION_FILE_DIR'] = './flask_session/'
app.config['SESSION_USE_SIGNER'] = True
Session(app)

db = SQLAlchemy(app)
login_manager = LoginManager(app)

# User model
class User(db.Model, UserMixin):...

# Paper model
class Paper(db.Model):...
```

Figure 5

3.3 Sprint 3

3.3.5 Sprint Retrospective

Table 3

	Sprint Retrospective		
Liked	**Learned**	**Lacked**	**Longed For**
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
The successful integration of front-end and back-end components was a highlight, demonstrating the power of collaborative efforts.	ained insights into effective API design, particularly around RESTful principles and handling error responses.	There was insufficient documentation for the backend services and API endpoints, making it challenging for new team members to onboard quickly.	Team members expressed a desire for better tools to facilitate API testing and debugging, such as Postman or Swagger.
The implementation of authentication and authorization mechanisms was well-received, reinforcing the importance of data security.	Improved debugging skills for resolving integration issues between the front end and back end.	The lack of automated testing for the backend led to some undetected issues during integration testing.	There was a longing for more time dedicated to comprehensive testing of the integrated system before the sprint concluded.

CHAPTER 4
RESULTS AND DISCUSSIONS

6.2 Project Outcomes

1. Unified Platform for Research and Development
   - Outcome: A fully integrated, user-friendly platform that combines features from code repositories (like GitHub) and research databases (like Google Scholar).
   - Impact: Researchers and developers can collaboratively manage projects, version control, and share academic contributions, reducing the need for multiple tools.
2. Comprehensive User and Project Management
   - Outcome: Features that allow users to create and manage personal profiles, collaborate on projects, and share their academic publications.
   - Impact: Users benefit from streamlined workflows for managing academic content alongside software projects, promoting interdisciplinary collaboration.
3. Secure and Robust Authentication System
   - Outcome: A reliable, secure authentication and authorization system with role-based access control, allowing users to securely access and manage their data.
   - Impact: Ensures data security and user privacy, giving users confidence that their sensitive information is protected.
4. Seamless Frontend-Backend Integration
   - Outcome: A well-integrated frontend and backend that offers a smooth user experience, with features for real-time data retrieval, project updates, and publication management.
   - Impact: Enhanced platform performance and responsiveness, allowing users to access information quickly and interact with the system without delays.
5. Centralized Repository for Academic and Code Contributions
   - Outcome: Users can store, organize, and access research papers, publications, and code projects in a single, centralized repository.

- Impact: Reduces administrative burdens and supports knowledge sharing by bringing all relevant materials together in one space, accessible for collaboration and reference.

6. Interdisciplinary Collaboration and Innovation
   - Outcome: An environment that fosters interdisciplinary research and software development collaborations, helping bridge gaps between academic and software development communities.
   - Impact: Drives innovation by enabling diverse fields to collaborate, potentially leading to new insights, tools, and advancements in science and engineering.
7. Enhanced Productivity and Workflow Efficiency
   - Outcome: Features that streamline the process of document uploads, project tracking, and publication sharing, minimizing redundant tasks.
   - Impact: Boosts user productivity, enabling researchers and developers to focus more on their core tasks rather than managing multiple platforms.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

The "Scholar Connect" platform has successfully achieved its primary objective of merging collaborative software development and academic research dissemination into a single, integrated environment. By combining the functionality of a code repository with features typical of academic databases, the platform offers a unique solution that supports interdisciplinary collaboration, secure data management, and streamlined workflows for researchers and developers alike. This project lays a robust foundation for further development, providing a seamless and productive environment where users can manage projects, track version histories, and share scholarly contributions in one place. Scholar Connect’s secure authentication and role-based access controls contribute to data privacy and user confidence, making it a dependable tool in the research and development community.

5.2 Future Enhancements

1. Enhanced Collaboration Features
    - Description: Introduce more advanced collaboration tools, such as real-time document editing, versioning for publications, and commenting or annotation features on research papers and code.
    - Benefit: Improves teamwork, enabling users to communicate and collaborate directly within the platform, which is essential for interdisciplinary research.
2. Subscription Feature
    - Description: Implement Subscription feature that allows express connections
    - Benefit: Allows subscribers to get faster response to increase the productivity
3. Integration with Popular Tools
    - Description: Expand compatibility with external tools like Google Scholar, ORCID, and GitLab, enabling users to import/export publications and code seamlessly.
    - Benefit: Enhances the platform's usability by allowing users to connect with widely used resources, making Scholar Connect an even more versatile and user-friendly ecosystem.
4. Advanced Data Analytics and Reporting

- Description: Add analytics dashboards and reporting tools to help users track project metrics, publication reach, and collaboration history.
- Benefit: Provides researchers and developers with insights into their projects' impact and productivity, supporting data-driven decision-making.

5. Improved Mobile Accessibility
    - Description: Develop a mobile-friendly interface or a dedicated mobile app to allow users access to their projects and publications on the go.
    - Benefit: Increases accessibility, enabling users to stay connected with their work anytime and anywhere, which is particularly useful for global collaboration.
6. Community-Building Features
    - Description: Implement features like user forums, interest groups, or a knowledge-sharing space to foster community interaction.
    - Benefit: Encourages knowledge exchange and networking among researchers and developers, creating a vibrant, supportive community around the platform.

References

1. D. Spinellis, “Version control systems,” IEEE Softw., vol. 22, no. 5, pp. 108–109, Sep. 2005.

2. A. Halevi, H. Moed, and J. Bar-Ilan, “Suitability of Google Scholar as a source of scientific information and as a source of data for scientific evaluation,” Rev. Inform. Sci., vol. 8, no. 3, pp. 100-108, Jul. 2007.

3. R. Ghosh and V. Vedam, “Collaborative Development with GitHub: The Definitive Guide,” O'Reilly Media, 2016.

4. Ziegler, A., et al. (2022). "Productivity assessment of neural code completion." *Proceedings of the 44th International Conference on Software Engineering*.

5. Imai, S. (2022). "Is GitHub Copilot a substitute for human pair-programming?" *Proceedings of the 30th ACM Conference on Computer-Supported Cooperative Work & Social Computing*.

6. Vaithilingam, P., Zhang, T., & Glassman, E.L. (2022). "Expectation vs. experience: Evaluating code generation tools." *Conference on Human Factors in Computing Systems*.

7. Nguyen, N. & Nadi, S. (2022). "An empirical evaluation of GitHub Copilot’s code suggestions." *Mining Software Repositories Conference (MSR)*.

8. Pearce, H., Ahmad, B., & Tan, B. (2022). "Asleep at the keyboard? Assessing GitHub Copilot’s code contributions." *IEEE Symposium on Security and Privacy (SP)*.

9. Sun, Z., et al. (2022). "CoProtector: Protecting open-source code from unauthorized training." *Proceedings of the ACM Web Conference*.

10. Wermelinger, M. (2023). "Using GitHub Copilot to solve simple programming problems." *SIGCSE Technical Symposium on Computer Science Education*.

11. Denny, P., Kumar, V., & Giacaman, N. (2023). "Conversing with Copilot: Exploring prompt engineering." *SIGCSE Technical Symposium on Computer Science Education*.

12. Finnie-Ansley, J., et al. (2023). "Testing OpenAI’s Codex on CS2 programming exercises." *Proceedings of ACM International Conference*.

13. Siddiq, M.L., et al. (2022). "SecurityEval dataset: Evaluating machine learning-based

code generation techniques." *ACM International Conference on Predictive Models in Software Engineering*.

14. Puryear, B., & Sprint, G. (2022). "GitHub Copilot in the classroom: Learning to code with AI assistance." *Journal of Computing Sciences in Colleges*.

15. PaperQA: Retrieval-Augmented Generative Agent for Scientific Research. (2023).

16. PuLID: Pure and Lightning ID Customization for Text-to-Image Generation. (2024).

17. Hi3D: Pursuing High-Resolution Image-to-3D Generation. (2024)

18. "Do Anything Now: Evaluating jailbreak prompts on large language models." (2023)

APPENDIX A

CODING

Frontend

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chat</title>
    <style>…
    </style>
</head>
<body>
    <h1>Chat with {{ recipient_user.username }}</h1>

    <div class="chat-container">
        <div class="messages" id="messages">
            {% for message in messages %}
                <div class="message {% if message.sender_id == session['user_id'] %}sent{% else %}received{% endif %}">
                    {{ message.message }}
                </div>
            {% endfor %}
        </div>
        <form method="POST" class="input-container">
            <input type="text" id="messageInput" name="message" placeholder="Type your message..." aria-label="Type your message" require
            <button type="submit">Send</button>
        </form>
    </div>
</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chat Partners</title>
    <style>
        body {...
        }
        h1 {...
        }
        .partner-list {...
        }
        .partner {...
        }
        .partner a {...
        }
        .partner a:hover {...
        }
    </style>
</head>
<body>
    <h1>Your Chat Partners</h1>

    <div class="partner-list">
        {% for partner in partners %}
            <div class="partner">
                <a href="{{ url_for('chat_inbox', recipient_id=partner.id) }}">{{ partner.username }}</a>
            </div>
        {% else %}
            <div>No chat partners found.</div>
        {% endfor %}
    </div>
</body>
</html>
```

Backend

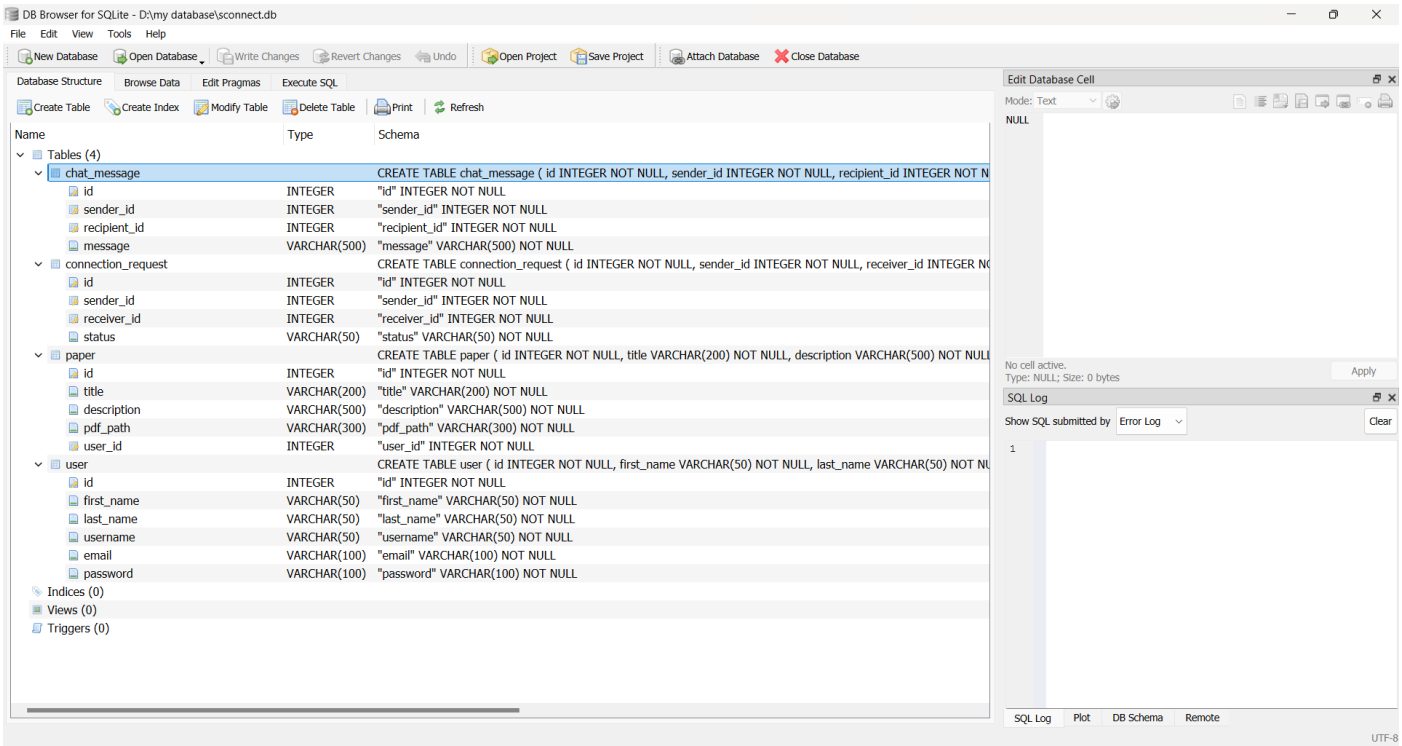
```python
class ChatMessage(db.Model):...

# Create the database tables
with app.app_context():...

# Global list for notifications
notifications = []

@login_manager.user_loader
def load_user(user_id):...

# Route: Landing Page
@app.route('/')
def index():...

# Route: Registration Page
@app.route('/register', methods=['GET', 'POST'])
def register():...

# Route: Login Page
@app.route('/login', methods=['GET', 'POST'])
def login():...

# Route: Dashboard
@app.route('/dashboard')
@login_required
def dashboard():...

# Route: Profile
@app.route('/profile')
@login_required
def profile():...
```

```python
@app.route('/logout')
@login_required
def logout():...

# Route: Upload a paper
@app.route('/upload', methods=['GET', 'POST'])
@login_required
def upload():...

# Route: Explore (List of papers)
@app.route('/explore')
def explore():...

# Route to delete paper
@app.route('/delete_paper/<int:paper_id>', methods=['POST'])
@login_required
def delete_paper(paper_id):...

# Route to handle collaboration request (now opens chat inbox)
@app.route('/collaborate/<int:paper_id>', methods=['POST'])
@login_required
def collaborate(paper_id):...

# Route to display chat partners
@app.route('/chats', methods=['GET'])
@login_required
def chats():...

# Route for individual chat inbox
@app.route('/chat/<int:recipient_id>', methods=['GET', 'POST'])
@login_required
def chat_inbox(recipient_id):...

# Route: Notifications
@app.route('/notifications_page')
```

```python
def chat_inbox(recipient_id):...

# Route: Notifications
@app.route('/notifications_page')
@login_required
def notifications_page():...

# Route: Modify submission
@app.route('/modify_submission/<int:paper_id>', methods=['GET', 'POST'])
@login_required
def modify_submission(paper_id):...

# Main entry point
if __name__ == '__main__':...
```

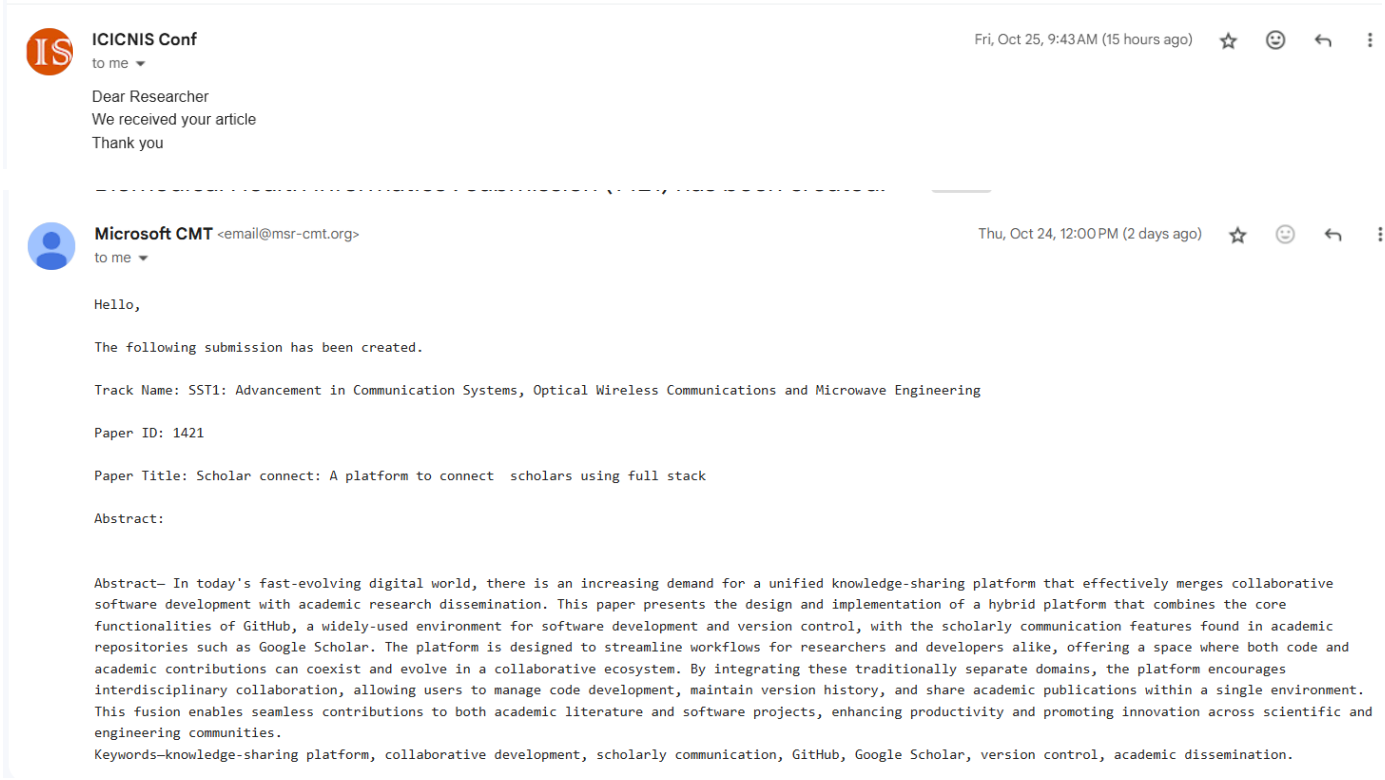
```python
from flask import Flask, render_template, request, redirect, url_for, flash, session
from flask_sqlalchemy import SQLAlchemy
from werkzeug.utils import secure_filename
import os
from flask_session import Session
from flask_login import LoginManager, UserMixin, login_required, current_user, login_user, logout_user
from flask_bcrypt import Bcrypt

app = Flask(__name__)
app.secret_key = 'b5a1f4a2c79dbf57b20f02d0e01f5c92'

# Initialize Bcrypt
bcrypt = Bcrypt(app)

# SQLite Database configuration
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///D:/my database/sconnect.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

# Server-side session configuration
app.config['SESSION_TYPE'] = 'filesystem'
app.config['SESSION_PERMANENT'] = False
app.config['SESSION_FILE_DIR'] = './flask_session/'
app.config['SESSION_USE_SIGNER'] = True
Session(app)

db = SQLAlchemy(app)
login_manager = LoginManager(app)

# User model
class User(db.Model, UserMixin):···

# Paper model
class Paper(db.Model):···
```

Database


DB Browser for SQLite - D:\my database\sconnect.db
File Edit View Tools Help
New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database Close Database
Database Structure Browse Data Edit Pragmas Execute SQL
Create Table Create Index Modify Table Delete Table Print Refresh
Name Type Schema
Tables (4)
chat_message CREATE TABLE chat_message (id INTEGER NOT NULL, sender_id INTEGER NOT NULL, recipient_id INTEGER NOT N
id INTEGER "id" INTEGER NOT NULL
sender_id INTEGER "sender_id" INTEGER NOT NULL
recipient_id INTEGER "recipient_id" INTEGER NOT NULL
message VARCHAR(500) "message" VARCHAR(500) NOT NULL
connection_request CREATE TABLE connection_request (id INTEGER NOT NULL, sender_id INTEGER NOT NULL, receiver_id INTEGER NO
id INTEGER "id" INTEGER NOT NULL
sender_id INTEGER "sender_id" INTEGER NOT NULL
receiver_id INTEGER "receiver_id" INTEGER NOT NULL
status VARCHAR(50) "status" VARCHAR(50) NOT NULL
paper CREATE TABLE paper (id INTEGER NOT NULL, title VARCHAR(200) NOT NULL, description VARCHAR(500) NOT NULL
id INTEGER "id" INTEGER NOT NULL
title VARCHAR(200) "title" VARCHAR(200) NOT NULL
description VARCHAR(500) "description" VARCHAR(500) NOT NULL
pdf_path VARCHAR(300) "pdf_path" VARCHAR(300) NOT NULL
user_id INTEGER "user_id" INTEGER NOT NULL
user CREATE TABLE user (id INTEGER NOT NULL, first_name VARCHAR(50) NOT NULL, last_name VARCHAR(50) NOT NU
id INTEGER "id" INTEGER NOT NULL
first_name VARCHAR(50) "first_name" VARCHAR(50) NOT NULL
last_name VARCHAR(50) "last_name" VARCHAR(50) NOT NULL
username VARCHAR(50) "username" VARCHAR(50) NOT NULL
email VARCHAR(100) "email" VARCHAR(100) NOT NULL
password VARCHAR(100) "password" VARCHAR(100) NOT NULL
Indices (0)
Views (0)
Triggers (0)
Edit Database Cell
Mode: Text
NULL
No cell active.
Type: NULL; Size: 0 bytes
Apply
SQL Log
Show SQL submitted by Error Log
Clear
1
SQL Log Plot DB Schema Remote
UTF-8

CONFERENCE PUBLICATION

We have submitted our paper for journaling and the proof is attached below



ICICNIS Conf
to me

Fri, Oct 25, 9:43 AM (15 hours ago)

Dear Researcher
We received your article
Thank you

Microsoft CMT <email@msr-cmt.org>
to me

Thu, Oct 24, 12:00 PM (2 days ago)

Hello,

The following submission has been created.

Track Name: SST1: Advancement in Communication Systems, Optical Wireless Communications and Microwave Engineering

Paper ID: 1421

Paper Title: Scholar connect: A platform to connect scholars using full stack

Abstract:

Abstract— In today's fast-evolving digital world, there is an increasing demand for a unified knowledge-sharing platform that effectively merges collaborative software development with academic research dissemination. This paper presents the design and implementation of a hybrid platform that combines the core functionalities of GitHub, a widely-used environment for software development and version control, with the scholarly communication features found in academic repositories such as Google Scholar. The platform is designed to streamline workflows for researchers and developers alike, offering a space where both code and academic contributions can coexist and evolve in a collaborative ecosystem. By integrating these traditionally separate domains, the platform encourages interdisciplinary collaboration, allowing users to manage code development, maintain version history, and share academic publications within a single environment. This fusion enables seamless contributions to both academic literature and software projects, enhancing productivity and promoting innovation across scientific and engineering communities.
Keywords—knowledge-sharing platform, collaborative development, scholarly communication, GitHub, Google Scholar, version control, academic dissemination.