

**ADEKUNLE AJASIN UNIVERSITY, AKUNGBA AKOKO**

**NAME: ADENIYI MATTHEW TEMITAYO**

**MATRIC NUMBER: 190404056**

**DEPARTMENT: COMPUTER SCIENCE**

**PROJECT TOPIC: DESIGN AND IMPLEMENTATION OF A WEB BASED REPOSITORY FOR UNDERGRADUATE PROJECTS**

**CHAPTER 1 & 2**

**PROJECT SUPERVISOR**

**MR AYEGBUSI A.O**.

**CERTIFICATION**

I, ADENIYI MATTHEW TEMITAYO. Hereby declare that this report has been carried out by me under the supervision of MR AYEGBUSI. It has not been presented for award of any degree in any institution. All sources of information are specifically acknowledged by means of reference.

Supervisor -----------------------

MR AYEGBUSI A.O.

Signature/Date

ADENIYI MATTHEW TEMITAYO -----------------------

Signature/Date

**CHAPTER ONE**

**INTRODUCTION**

**1.1 Background of Study**

Storage facilities can store learning materials like a library stores books or information about them.

They help in finding, sharing, and reusing learning materials (Richards, McGreal, and Friesen, 2002).The Computer Science Department at Adekunle Ajasin University Akungba Akoko is in the process of developing an online repository. This repository aims to provide a user-friendly interface, granting easy access to undergraduate projects completed by students of the department. Users will be able to conveniently retrieve project abstracts and relevant details such as the student's name, the year of project submission, the project topic, and the project itself. The development of this repository is currently underway to enhance accessibility and dissemination of academic works within the department.

Currently, accessing physical project archives in the department is cumbersome and inefficient. Physical records are prone to damage, loss, and are not easily searchable, creating significant barriers for students and faculty who need to reference past works. The proposed digital repository aims to address these issues by providing a robust, searchable, and easily accessible platform for storing and retrieving academic projects.

Digital repositories have become essential in modern academic environments. They not only preserve academic outputs but also provide broader access to research findings, fostering collaboration and innovation. Implementing a digital repository for the Computer Science Department will address current limitations in accessing physical project archives and ensure the preservation of academic works against potential damage or loss. Furthermore, it aligns with global trends in academia where digital repositories are leveraged to promote open access to research outputs, thereby contributing to the global body of knowledge.

The repository will significantly increase the visibility and impact of research conducted within the institution. By making undergraduate projects accessible online, the repository will allow current and prospective students, faculty, and external researchers to view and build upon existing work. This increased visibility can lead to enhanced collaboration, citations, and overall academic contribution, strengthening the institution's reputation in the research community.

Moreover, the establishment of a digital repository can streamline the administrative process associated with project submission, review, and storage. It will facilitate efficient management of academic records and ensure that projects are systematically archived and easily retrievable. This not only supports the academic mission of the Computer Science Department but also aids in administrative efficiency and resource management, providing a robust platform for the long-term sustainability of academic documentation and research outputs.

The repository will be developed using robust technologies to ensure security, scalability, and user-friendliness. It will feature an intuitive user interface that allows users to browse the project catalog and search by student name, project topic, or year of project submission. This will help potential graduates of the department to know the topics that have been researched, the areas of concentration, and the extent of each research project.

However, the development and implementation of the repository will not be without challenges. Potential obstacles include ensuring data security, maintaining user privacy, and integrating the repository with existing institutional systems. To address these challenges, the project will employ best practices in web development and data management, ensuring robust encryption, user authentication, and regular system updates.

Similar successful projects at other institutions, such as MIT's DSpace demonstrate the viability and benefits of digital repositories. Learning from these examples, the Computer Science Department at Adekunle Ajasin University will adopt best practices to ensure the repository meets the highest standards of academic and technical excellence.

In conclusion, the development of a web-based repository for undergraduate projects at the Computer Science Department of Adekunle Ajasin University Akungba Akoko is a critical initiative. It will enhance the accessibility, visibility, and impact of academic work, streamline administrative processes, and align with global academic trends. By addressing current limitations and leveraging modern technologies, the repository will provide a sustainable platform for preserving and disseminating academic knowledge.

**1.2 Statement of the Problem**

Projects already submitted to the Computer science department cannot be accessed anywhere and at any time without having to come to the school building and reaching the office in charge. In case of fire incidence or water, the copies of the projects curated by the school could be damaged or totally lost. Searching of projects is very hectic and slow due to the quantity of projects and limited human effort. In addition, access to projects can be delayed due to the official procedures and protocols involved in the process. Therefore, this work is to develop a repository to solve this problem.

**1.3 Motivation of study**

The motivation behind this study stems from the need to enhance the accessibility and preservation of academic projects within the Computer Science department. A web-based repository will eliminate physical and procedural barriers to accessing projects. This digital solution is inspired by the goal of providing a robust platform for academic resources, ensuring their longevity and wider availability.

**1.4 Aim-and-Objectives-of-the-Study**

AIM: The aim of this study is to develop a web-based repository for undergraduate projects in the Computer science department. This will be achieved through the following specific objectives:

1. To design a web-based repository for undergraduates in the Computer science department,

Adekunle Ajasin University,Akungba Akoko.

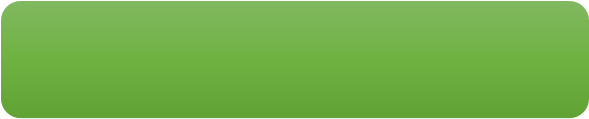
1. To implement the design outlined in (i) using PHP and MySQL.

**1.5 Methodology**

The development of the web-based repository will follow a structured methodology. This includes:

Start

Requirement Analysis



System Design

Implementation

Testing

Deployment

Maintenance

End

**1.6 Significance of the Study**

A key function of this research is that this repository provides a unique storage identifier for every object, therefore making it feasible with the aid of the search module incorporation into the web-based repository application to efficiently search, discover and retrieve objects (projects) from its database**.** Both staff and student will benefit from this feature in that one can easily filter or search for a project by the project name, student name or year. The office in charge of keeping and handling undergraduate projects will now have better support for storage of graduates’ projects and easy retrieval on request.

The Computer science department repository could also serve as an educational environment for many internet users and researchers. It will encourage researchers to take up further research on a subject matter (like a project) and thereby add to the body of knowledge. This repository also covers authors of the projects and because it clearly attributes the student name (author) to its content.

**1.7 Expected Contribution to Knowledge**

This study is expected to contribute to knowledge by providing a systematic and efficient way of managing and accessing undergraduate projects. It will enhance the academic research process by making past projects readily available, thereby promoting continuity in research. Additionally, it will serve as a model for other departments and institutions seeking to implement similar digital repositories.

**1.8 Scope and Limitation study**

This study focuses on undergraduate projects of the Computer Science Department at Adekunle Ajasin University, Akungba Akoko Each project will include the project topic, student name, abstract, year of submission, and the full project itself. Users will be able to search the project catalog by project topic, student name, keyword, or year of submission.

The repository will be accessible online and will require an active internet connection and a browser. It cannot be accessed offline unless installed on a computer with an active server application and a MySQL database application. Projects cannot be downloaded for offline use.

**1.9 Definition of Terms**

**Learning Objects:** In the context of this research, a learning-object-is-a-resource with a clear educational-application. It is in digital form – examples: a Microsoft Word Document or a PDF document.

**Catalog:** This research applies the word ‘catalog’ as a container that presents all the Computer Science Department projects logged/stored in an organized tabular format, showing relevant information about each project. It can be compared to a library catalog containing all the information about books – their titles, author, number of pages, published year, ISBN etc.

**Repository:** A repository (in this case; Computer Science Department repository) is an archive, storehouse or container that allows storage, cataloging, accessing all Computer Science Department undergraduate projects and thesis, and viewing each object’s content.

**Identifier:** An identifier is simply a unique identity attached to one object, it aids referencing such object directly when its identifier is requested. An identifier can be a serial number, identity number (also written as ‘ID’) or an index number.

**Object:** An object in the context of this research refers to a project record returned by a database – queried using an Object Oriented Programming (OOP) approach. An object (i.e. a project record) returns or contains the project topic, student name, abstract, year of project submission and the date and time, the object record was created.

**Database:** A database is a repository that allows storage, retrieval and manipulation of data. A database can be used efficiently with the aid of a Database Management System (DBMS) – a set of tools that allow storage, access, retrieval and maintenance of data stored in a database, examples of DBMS include XAMPP, WAMP, LAMP and AppServ to mention a few. A database understands only the Structured Query Language (SQL) which is used to communicate with it, popular databases used by well-known Information Technology (IT) firms include are MySQL, Oracle and Django DB.

**Web-Browser:** A web-browser,-internet-browser-or-browser-is-software program that interprets the codes written in markup languages in graphic and visual (like images, text, audio or animation) form. This allows users to easily request and access a website or to search for information through a search engine (either Google, Ask, Bing or AOL among others).

**Keyword:** A phrase or just one word that is used to search for a certain result or set of results. **User:** A user is system, application, request or person that can use a computer or software to perform a specific task.

**Metadata:** A set of data that describes and gives information about other data.

**Dataset:** A collection of related game plans of information that is made out of disengaged segments however can be controlled as a unit by a PC.

**Relational Database:** A database composed to see association among relations of information delineates tables from which data can be gotten to or reassembled in different courses without rearranging the database tables.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Historical Development of Digital Repositories**

The World Wide Web was initially used by researchers in various fields, including physics, aviation, and even social sciences, to share their findings more quickly than traditional paper-based journals allowed. However, it was not until the mid-1990s, with the rise of user-friendly interfaces and widespread internet access in academic settings that dedicated online repositories truly flourished. These subject-based repositories allowed researchers from diverse disciplines to share pre-prints of articles before publication and were primarily focused on fields like material science, biology, and chemistry, where rapid dissemination of results is crucial (English, 2010).

Institutional repositories (IRs) can play a crucial role in promoting open access (OA) by comprehensively collecting the intellectual output of universities and research institutions. This goes beyond just "green" OA, which includes peer-reviewed journal articles, to encompass a wider range of scholarly materials. IRs can archive student theses, data sets, working papers, blog posts, and more. By doing this, IRs become essential platforms for reaching a broader audience, including researchers, students, and the public. They can also amplify new voices in academic discourse and foster collaboration (Williams, Pope and Lucero, 2014).

The rise of IRs (Institutional repositories) in the mid-2000s stemmed from several converging factors. First, advancements in digital publishing technologies and widespread internet access allowed libraries to enter the dissemination space at a significantly lower cost compared to traditional academic presses. Second, the storage and access costs for large volumes of scholarly content dropped dramatically. Third, universities began to grapple with the rising costs of traditional print journals due to factors like peer review, copyright, and production models that favored physical distribution over digital access. Finally, the instability of the scholarly publishing industry heightened concerns about the long-term preservation of digital research materials, a concern that remains relevant today. In response to these circumstances, universities started implementing IRs, which could facilitate a collaborative, open access, and globally accessible model for disseminating research, while simultaneously ensuring the curation and preservation of an institution's entire scholarly output (English, 2010).

**2.2 The Computer Science Department Repository**

Traditionally, scholarly research has been disseminated primarily through academic journals and conferences. However, the high costs associated with traditional publishing can limit access to valuable research outputs. The Open Access (OA) movement aims to make scholarly research freely available online, democratizing access to knowledge and accelerating scientific progress. Institutional repositories are a key component of the OA movement, serving as digital collections of the intellectual output created within a university or research institution. These repositories typically house faculty research articles, theses, and other scholarly works, providing open access to a wider audience than traditional publishing channels.

The computer science department repository exemplifies this concept. It serves as a central hub for the department’s scholarly output, providing open access to research by faculty and students.

This not only increases the visibility and impact of the department’s research but also fosters collaboration and knowledge sharing within the broader computer science community. Researchers and students around the world can easily access and utilize valuable resources, leading to faster advancements in the field.

The principles of Open Access are rooted in the belief that knowledge should be shared freely and openly, without barriers. This philosophy aligns with the academic community’s dedication to the advancement of science and education. However, the implementation of OA (Open Access) is not without its challenges. Issues such as copyright compliance, data preservation, and equitable access across different regions must be addressed to fully realize the benefits of OA.

Further research reveals that institutional repositories significantly enhance the discoverability of research outputs. Studies have shown that articles deposited in OA repositories receive a higher citation count compared to those available only through subscription-based journals.

While the departmental repository has successfully increased the accessibility of research, it is crucial to critically assess its effectiveness. Comparisons with other repositories, such as the MIT OpenCourseWare or the arXiv preprint server, may provide insights into best practices and areas for improvement.

Setting up an Open access (OA) repository involves careful planning and consideration of various factors. Technical infrastructure, long-term funding, and clear policies on intellectual property rights are essential components that ensure the sustainability of the repository.

Looking ahead, the evolution of OA repositories may include integration with emerging technologies like block chain for enhanced security and transparency. Additionally, the rise of data-driven research necessitates the development of advanced data management capabilities within these repositories.

Peake (2012) elucidates that institutional repositories adhere to a globally recognized set of technical standards, ensuring that they uniformly disclose essential information such as author names, organizational affiliations, publication dates, and article titles. This standardization is facilitated by the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), which enables the contents of all repositories to be indexed by internet search engines like Google and Google Scholar. Consequently, these online databases become rich sources of freely accessible global research. The ubiquity of search engines means that anyone with internet access can discover scholarly works or datasets housed within an academic or research institution’s repository through a simple web search.

In addition to search engine discovery, repositories are accessed through more direct methods. Users may navigate to a specific repository to find works by a particular researcher affiliated with the institution, or they may follow links from a researcher’s personal website or academic journal.

While these “referrals” are noteworthy, the most common avenue for users to encounter a repository is through a general internet search engine query.

In the context of the computer science repository, each object within the repository is associated with a set of metadata that includes the author’s name(s), organizational affiliation (represented by an undergraduate number), date of publication, title of the study, abstract, and supervisor’s name. This standardized metadata is essential for all objects housed in the repository, ensuring uniformity and facilitating ease of access and retrieval.

Moreover, various web search engines, such as Google, Yahoo, AOL, Bing, and Ask, index the contents of the computer science repository. This indexing allows anyone with internet access to locate information about a project within the computer science repository through a web search.

The repository’s design enables users to conduct targeted searches for specific project works, filtering results by the student’s name (graduate), the supervisor’s name at the time of the study, or the year the work was submitted to the faculty. Such search capabilities enhance the discoverability of scholarly works and contribute to the broader dissemination of academic knowledge.

**2.3 Types of Repositories**

A repository is a central location that stores, manages, and allows access to digital information. This can include things like documents, code, images, and multimedia files. Users can find, retrieve, and even display or use these digital objects within the repository.

Open Access Repositories (OARs) are digital databases that provide free access to scholarly literature and research outputs for anyone. Unlike traditional repositories, OARs do not limit users' ability to read, download, and sometimes even reuse the content, following the terms of specific licenses. As knowledge creation increasingly shifts to the digital realm, repositories are developing innovative methods to archive, preserve, and provide access to the data and materials that underpin peer-reviewed research. A growing number of research funders now mandate Open Access (OA) publication of research outputs, including the data underlying the findings. This promotes transparency and reproducibility, allowing other researchers to verify results, compare them with their own findings, and potentially reuse the data to generate new knowledge and discoveries. Datasets can come in various formats, including spreadsheets, images, audio files, video files, design software outputs, diagrams, and charts. Additionally, complex objects like datasets combined with related visualizations (e.g., weather data with satellite images) are also being deposited in repositories. (Source: Open-Access Repositories, 2015)

A growing number of hierarchical repositories now include books or book chapters alongside other scholarly content. While books are traditionally written for commercial gain (through royalty deals), authors might be hesitant to make them entirely free in open access repositories. However, even in these cases, preserving bibliographic information (like title, author, and publisher details) in the repository is crucial. The content itself can be "hidden" from public view. This information ensures the book is indexed by search engines and included in institutional evaluation processes. Interestingly, research suggests that even when the full book content is available in a repository, sales can actually increase. This is likely because the increased visibility within the repository brings the book to the attention of a wider audience who might then be more likely to purchase it if it is relevant to their needs. This effect is similar to what Amazon achieves with its "Look Inside" feature" (Swan, 2009).

In addition to the previously mentioned content types, the computer science repository also includes undergraduate research outputs such as titles, author names (students), abstracts, and supervisor names. The visibility of an item in the repository is determined by its view count. Frequently viewed objects rank higher in search results and appear at the top of the repository listings. For items with the same or no views, the computer science repository employs a reverse chronological order, showcasing newly submitted works at the top of the list.

“Why Open-Access? The term "Open Access" embodies its core principle: the unrestricted access to information for everyone, regardless of affiliation, education, race, ethnicity, religion, nationality, ancestry, sexual orientation, age, disability, legal status, or any other grouping. It represents the collective effort to make knowledge an open resource, fostering the advancement of human progress. (English, 2010).

Why Build a Digital-Repository? As educators, we have a commitment, both individually and as an institution, to ensure our scholarly work is widely accessible and preserved for the future. This valuable material represents a significant contribution to the global academic discourse. A digital repository allows us to achieve this by facilitating broad dissemination of our research, thereby enhancing the visibility and reputation of both ourselves and the college within the academic community. It also serves to demonstrate the broader social and economic impact of our scholarly endeavors. Currently, much of this valuable scholarly output exists in an unorganized and unprotected state. It's largely inaccessible outside the institution's walls and not readily discoverable by search engines, which is where most researchers begin their investigations. Without a systematic approach to collecting and managing these resources, their discoverability and accessibility over time cannot be guaranteed. Therefore, a digital repository is essential to create a central hub for organizing, preserving, and showcasing this important work. (English, 2010)

An Institutional Digital Repository (IDR), simply put, is a web-based archive for collecting, preserving, and providing access to the digital scholarly output of an institution, particularly a research institution. As Thorat and Pakil (2013) define it, an Institutional Repository (IR) acts as a digital platform where a university community's scholarly work is made openly accessible and securely stored for its creators.

An Institutional Digital Repository (IDR) caters to a wide range of scholarly materials. It can include traditionally published content like journal articles, peer-reviewed research papers, and proposals. Additionally, it can house unique digital resources created by the academic community, such as datasets for scientific research, course presentations, or other educational materials.

These institutional repositories are typically accessible online and open to the public. This is in contrast to most journal articles, which are often restricted to subscribers and not easily found by search engines like Google. Conversely, research papers deposited in an IDR are freely available to everyone (they are Open Access or OA) and can be discovered through regular web searches.

Well-known open-source software platforms like DSpace, EPrints, and Bepress are commonly used to build and manage institutional repositories (Institutional Repositories, 2014).

1. The four key purposes for having associate institutional repository are:
2. To create world visibility for associate institution's critical appraisal.
3. To collect content in associate passing single location.
4. To provide open access to institutional analysis output by self-archiving it.
5. To store and preserve different institutional soft-copy assets, moreover as unpublished then again generally easy to lose writing (for instance, field reports).

Institutional repositories share a close connection with the concept of a modern library. Both fulfill the core function of collecting, preserving, organizing, curating, and providing access to information. Just as traditional libraries manage physical materials like books and journals, institutional repositories handle digital scholarly content. This digital content can include research articles, datasets, multimedia files, and other valuable academic resources. (Institutional Repository, 2014).

Several tools exist today that map the global landscape of open access digital repositories. One such initiative is "Directory of Open Access Repositories" (OpenDOAR), managed by the Securing a Hybrid Environment for Research Preservation and Access (SHERPA) project. OpenDOAR functions as both a directory and a data source for another tool called "Store66." Data from OpenDOAR, since 2007, indicates that the most popular software platforms used to build institutional repositories are EPrints, and DSpace (Dora and Maharana, 2012).

**2.3.1 DSpace**

DSpace is a popular open-source software tool specifically designed for managing digital resources. It is mostly used by research universities as a platform for building institutional repositories (IRs). DSpace provides features for organizing and preserving a wide variety of digital content, including books, dissertations, 3D models, images, videos, research datasets, and more.

Information within DSpace is organized into collections, which group related items together. Notably, DSpace allows researchers to share their work with a global audience in a way that was previously challenging or impossible (Tansley, Smith & Harford, 2005).

Beyond its core functionalities, DSpace can also be used for digital preservation initiatives.

Developed in 2002 by the Hewlett-Packard-Massachusetts Institute of Technology Alliance (HPMIT), DSpace has been adopted by over 240 institutions worldwide. These institutions range from large universities to smaller teaching colleges, social organizations, and research centers. A key advantage of DSpace is its licensing. It is distributed under a Berkeley-Software Distribution (BSD) license, which allows institutions to customize or even extend the software's functionalities to meet their specific needs (DSpace, 2008).

The DSpace development team follows a structure similar to the Apache Foundation's model. This model consists of a user community, which includes a group of developers. Some of these developers contribute directly to the core codebase. Their contributions are then incorporated into the official release overseen by a central group of committers. These committers ensure that the code adheres to the established guidelines outlined in the developer documentation. This process helps maintain consistency and control over the direction of DSpace's development, ultimately benefiting the entire user community (DSpace, 2008).

DSpace is built using Java and Java Server Pages (JSP), leveraging the Java Servlet API for functionality. It relies on a relational database for data storage, with support for PostgreSQL and Oracle. DSpace primarily provides access to its content through a user-friendly web interface. However, it also adheres to the Open Archives Initiative Protocol for Metadata

Harvesting (OAI-PMH) version 2.0, allowing for interoperability with other systems.

Additionally, DSpace can create and manage Metadata Encoding and Transmission Standard (METS) packages, a standard format for preserving digital objects. Future versions are likely to see increased adoption of web services and potential changes to the user interface. One potential security challenge associated with DSpace is its ability to handle a wide variety of file formats and types. While this versatility makes DSpace a universal platform, it also introduces potential vulnerabilities. Since DSpace supports numerous formats, ensuring comprehensive security for the entire software becomes a complex and challenging task. Another challenge presented by DSpace's diverse file format support is its storage needs. Accommodating a wide variety of file types can lead to significant storage requirements. Large, expensive storage media with robust maintenance practices become necessary. In some cases, an entire data center, a dedicated facility designed for large-scale data storage, might be required to house all the digital objects. The Computer Science Department repository focuses on storing metadata, which is descriptive information about the scholarly resources. Unlike traditional repositories, it does not allow users to upload files directly. This characteristic simplifies security measures for the repository. Since there is no user-generated content, the risk of malicious file uploads is eliminated. The repository remains publicly accessible for research purposes, but users are granted read-only permission, restricting their ability to modify or delete information.

**2.3.2 EPrints**

“EPrints is a freely available open-source software platform designed to create digital repositories. These repositories specifically cater to open access, meaning they provide free online access to scholarly content. EPrints ensures compatibility with the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), which facilitates the sharing of information between repositories. While EPrints offers some functionalities similar to document management systems, its core purpose lies in supporting institutional archives and scholarly journals (Gumpenberger, 2009).

EPrints quickly became a leading free, open-source software for building Open Access Institutional Repositories (IRs), inspiring similar developments. Version 3, launched in January 2007 at the Open Repositories Conference, was praised for its enhanced administrative capabilities and flexibility for repository managers and technical staff (EPrints, 2015)

EPrints is a web-based application built on the LAMP architecture (Linux, Apache, MySQL, PHP). However, unlike the standard LAMP stack, EPrints is primarily written in Perl instead of PHP. It has been successfully run on various operating systems, including Linux, Solaris, and macOS. Version 3 of the software introduced a plugin architecture based on Perl for functionalities like data import/export, content transformation (facilitating search engine indexing), and user interface customization. Setting up an EPrints repository involves modifying configuration files written in Perl or Extensible Markup Language (XML). The user interface of an EPrints repository is primarily built using HyperText Markup Language (HTML) formats, templates, and inline images. While EPrints is initially released in English, it can be localized into other languages through redistributable, language-specific XML configuration files. Currently supported languages include, but are not limited to, German, French, Italian, Japanese, and Spanish.

While EPrints offers more functionality than the Computer Science Department repository, it has a steeper learning curve, supporting up to 10 languages and a command-line interface for advanced users. This can be a barrier for some administrators. In contrast, the Computer Science Department repository is easier to use and does not require command-line knowledge. EPrints accommodates a broader range of content, including journals and educational materials, whereas the Computer

Science Department repository focuses specifically on undergraduate projects from the School of

Information and Communication Technology.

**2.4 Merits and Demerits of Digital Repositories**

A. Merits: Digital Repositories are of many advantages, a few of that are:

1. **Increased Exposure:** Repositories make your research readily available to a global audience, potentially leading to more citations. Studies have shown that openly accessible publications tend to be cited more frequently.
2. **Universal Access:** Unlike journal articles limited to subscribers, repository content is accessible to anyone, significantly increasing the public value of research.
3. **Enhanced Discoverability:** Repositories leverage search engine services like Google, Google Scholar, and OCLC, allowing users to easily find relevant content within the repository.
4. **New Research Techniques:** Digital repositories facilitate new research methods such as data mining, text analysis, and visualization, enabling researchers to uncover hidden patterns and relationships within the data.
5. **Persistent Access:** Deposited works in repositories receive persistent URLs (permanent reference links) that never change, eliminating the issue of dead links. Unlike content on personal websites, repository materials remain accessible over time.
6. **Long-Term Preservation:** Digital repositories are managed by libraries committed to long-term access and preservation of the collection. Libraries ensure ongoing maintenance and data backups to safeguard the information.
7. **Rich Content Variety:** Digital repositories extend beyond journal articles, encompassing diverse materials like conference proceedings, images, and other research data. This allows for broader dissemination and access to a wider range of scholarly outputs. B. Demerits: Digital repositories even have their disadvantages, a number of that are:
8. **Version Control Issues:** Multiple versions of the same material can exist online, leading to confusion for researchers seeking the most accurate and current information. Effective version control strategies within repositories are crucial.
9. **Plagiarism Concerns:** Some authors hesitate to deposit their work in repositories due to copyright infringement risks. Clear copyright and licensing policies implemented by repositories can help address these concerns.
10. **Long-Term Preservation Costs:** Maintaining and ensuring the long-term accessibility of digital content requires ongoing investments in storage, security, and data management. Institutions need to develop sustainable funding models for repository upkeep.
11. **Potential for Substandard Content:** The ease of self-archiving can lead to authors depositing unreviewed or incomplete work. Implementing quality control mechanisms within the submission process can help mitigate this issue.

**2.5 Review of Related Literature**

* Abrizah (2009) emphasized effective Institutional Repository (IR) management, highlighting steps like understanding IR software, promoting its use, and educating faculty on self-archiving. Using qualitative analysis, Abrizah found challenges such as low faculty awareness of open access (OA) and the need for institutional mandates, addressing these with workshops and informational sessions.
* Thorat and Pakil (2013) discussed libraries’ role in shaping IRs, proposing a web forum for feedback. Analyzing successful global IRs through case studies and interviews, they identified challenges in securing long-term funding and sustainability, addressing these via community engagement and collaborative platforms.
* Bull and Eden (2014) stressed educating library staff on scholarly communication, including Open Access and copyright issues. Through surveys and training program assessments, they developed staff training programs but faced challenges in content focus and implementation, using interactive modules and continuous feedback to mitigate these issues.
* Rhaman and Mezbah-ul-Islam (2014) examined IRs in Bangladesh, identifying best practices and guidelines through surveys and comparative analysis. They highlighted the potential of IRs to overcome traditional barriers but noted the need for tailored local strategies, proposing customized guidelines to address these challenges.
* Chen and Kim (2017) examined integrating social networking features into web-based repositories to foster collaboration. They evaluated the effectiveness of these features and faced balancing challenges, addressed through pilot testing and user feedback.
* Garcia and Patel (2018) explored enhancing user engagement with web-based repositories through user behavior analysis and design principles. They identified factors influencing interaction and proposed design improvements, addressing engagement level and design implementation challenges with behavior tracking and iterative adjustments.
* Wang and Li (2019) investigated web-based repositories’ impact on knowledge dissemination, using usage data analysis and surveys. They highlighted repositories’ role in promoting open access but faced data collection challenges, addressed with analytical tools and user incentives.
* Clark and Nguyen (2020) presented a case study on showcasing undergraduate research via web-based repositories, using case study and impact analysis. They customized repositories for undergraduates and measured learning outcomes, addressing customization and impact measurement challenges with tailored solutions and surveys.
* Taylor and Martinez (2021) shared best practices for undergraduate capstone project repositories, using case study analysis. They faced platform selection and submission management challenges, addressed through platform evaluation and streamlined processes.
* Brown and Rodriguez (2022) tackled the development of web-based repositories for undergraduate research, focusing on metadata standards and faculty buy-in. Through metadata analysis and surveys, they developed standards and conducted workshops, overcoming challenges with collaborative standard-setting and advocacy.
* Johnson and Smith (2023) focused on web-based repositories for undergraduate projects, using user-centered design and usability testing. They faced challenges in student engagement and workflow integration, addressed through iterative design and user feedback.

**2.6 Summary of Review**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S/N | **TITLE** | **AIM** | **METHODOLOGY** | **RESULT** | **LIMITATION** |
| 1 | Effective IR  Management  (Abrizah,2009) | To outline key steps for  managing  Institutional  Repositories  (IRs) | Review of existing practices and survey of faculty regarding IR awareness and practices | Identified essential IR  management  practices; low faculty awareness of OA publishing | Need for institutional mandates and strategies to increase faculty participation |
| 2 | Influence of  IRs on  Scholarly  Communication (Thorat and Pakil, 2013) | To discuss the role of IRs in the scholarly communication cycle | Examination of global IR implementations and a proposed web forum for feedback | Highlighted the influence of IRs on scholarly communication proposed a web forum for feedback | Securing long-term funding and ensuring IR sustainability |
| 3 | Promoting  Scholarly  Communication (Bull and Eden, 2014) | To emphasize the importance of educating library staff on scholarly communication best practices | Analysis of educational  programs and  staff interactions with faculty and students | Identified key issues for staff education on OA, copyright, and publishing challenges | Challenges in  implementing effective staff education programs |
| 4 | Status of IRs in  Bangladesh  (Rhaman and  Mezbah-ulIslam, 2014) | To investigate the status and development of IRs in  Bangladesh | Review of existing IR  practices in  Bangladesh and analysis of secondary sources and surveys | Identified best practices for IR development in Bangladesh | Need for tailored strategies to address local challenges |
| 5 | Integrating  Social  Networking in  Web-Based  Repositories  (Chen and  Kim, 2017) | To examine the integration of social features in repositories for collaboration | Evaluation of social networking features and their impact on user engagement and collaboration | Enhanced user engagement and collaboration through social features | Ensuring  effective implementation  of social features and measuring their impact |
| 6 | Enhancing  User  Engagement with Web-Based  Repositories  (Garcia and  Patel, 2018) | To explore strategies for enhancing user engagement with web-based repositories | Analysis of user interaction factors and design principles for repository platforms | Identified factors for optimizing user experience; promoted content discovery | Technical and design challenges in maintaining high user engagement |
| 7 | Impact of  Web-Based Repositories on Knowledge Dissemination (Wang and Li,  2019) | To investigate the role of web-based  repositories in academic knowledge dissemination | Usage pattern analysis of faculty and student interactions with the system | Enhanced knowledge dissemination; promoted open access | User engagement strategies and long-term sustainability |
| 8 | Utilizing Web-  Based  Repositories for  Undergraduate  Research  (Clark and  Nguyen, 2020) | To showcase undergraduate research through web based repositories | Case study of repository customization and impact measurement | Customized repository to meet needs; measured impact on learning outcomes | Measuring long-term impact and user satisfaction |
| 9 | Best Practices for Web-Based Repositories  (Taylor and  Martinez,  2021) | To share lessons and best practices in  implementing repositories for capstone projects | Case study analysis of platform identification and project submission management | Identified appropriate platforms; managed  project submissions | Technical challenges in platform customization |
| 10 | Development and Evaluation of Web-Based Repositories  (Brown and  Rodriguez,  2022) | To discuss challenges and opportunities in web-based repositories for undergraduate research | Evaluation of metadata standards and faculty buy-in for promoting student participation | Addressed issues in metadata standards; promoted faculty buy-in | Ensuring consistent student participation and engagement |
| 11 | Development of Web-Based Repositories for Undergraduate  Projects  (Johnson and  Smith, 2023) | To develop and implement web-based repositories for undergraduate projects | Design and analysis of a repository system with a focus on user-friendly interfaces and scalability | Highlighted the importance of user-friendly design promoted student engagement | Challenges in integrating the repository into existing academic workflows |

### CHAPTER THREE

### SYSTEM ANALYSIS AND DESIGN

**3.1 Analysis of the current System**

The Computer Science Department at Adekunle Ajasin University Akungba Akoko currently keeps physical copies of student projects with the departmental secretary. A list with each student’s name, their project topic, and the year it was completed is kept on paper. This makes it difficult and time-consuming to find a specific project, and the records can be damaged by things like rodents or other natural causes. Overall, the process of storing these projects is done manually, which can lead to delays and possible loss of important information.

**3.2 Limitations of the existing System**

The current manual system of storing projects presents several challenges:

1. **Finding records:** It takes a lot of time and effort to find specific past records because everything is done manually.
2. **Losing** **records**: Records can easily get lost when they are moved from one place to another.
3. **Limited access:** Since records are only on paper and kept in one location, they are not easily accessible. Even if someone needs basic information, they have to go through a long process of requesting it, which takes time and is inefficient.
4. **Damage** **from** **disasters**: These records are at risk of being destroyed by things like floods or fires, and if they are lost, it will take extra resources to recover them, if possible.

**3.3 Justification of the new System**

The proposed Computer Science Department repository can be accessed online from anywhere in the world through its website. This is a big improvement for the department, as it will reduce the need for the lengthy formal processes required to get basic project information. It uses a fast and simple search system, and the design is user-friendly and easy to navigate. The repository provides basic details for each project, like the project title, year of submission, student’s name, and supervisor’s name. Since it is hosted online, the risk of losing data is very low. Adding new projects to the system is also much easier compared to the current manual process.

**3.4 Methodology**

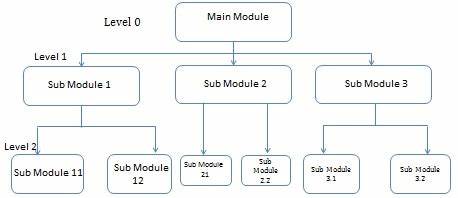
The proposed Computer Science repository is developed using a modular, top-down design approach shown in figure 3.1. This method starts with the overall system design and then breaks it down into smaller components, allowing for better clarity and ease of development. The top-down design ensures that the system is built from the larger elements first, gradually focusing on more detailed sub-systems.

The repository is a web-based client-server system, where the user interface (client), data storage, and information access are developed to function seamlessly together. The database for the system is implemented using XAMPP, which includes Apache (the server), MySQL (the database), and PHP.

Apache operates as the server, while MySQL provides the database for storing project information, with phpMyAdmin offering an easy-to-use interface for database management.

MySQL, as a relational database, processes data efficiently and securely. It includes features like encrypted access passwords, ensuring that only authorized users can access the database. Since MySQL is open source, it is both free and adaptable to specific needs.

The data structure for the Computer Science repository is object-oriented, meaning that each project stored in the database is treated as an object with its own attributes, such as project title, year of submission, and student details. This modular approach makes the system easier to maintain, allowing for quick testing, debugging, and future updates. The user interface is designed to be user-friendly, enabling smooth interaction between users and the repository.



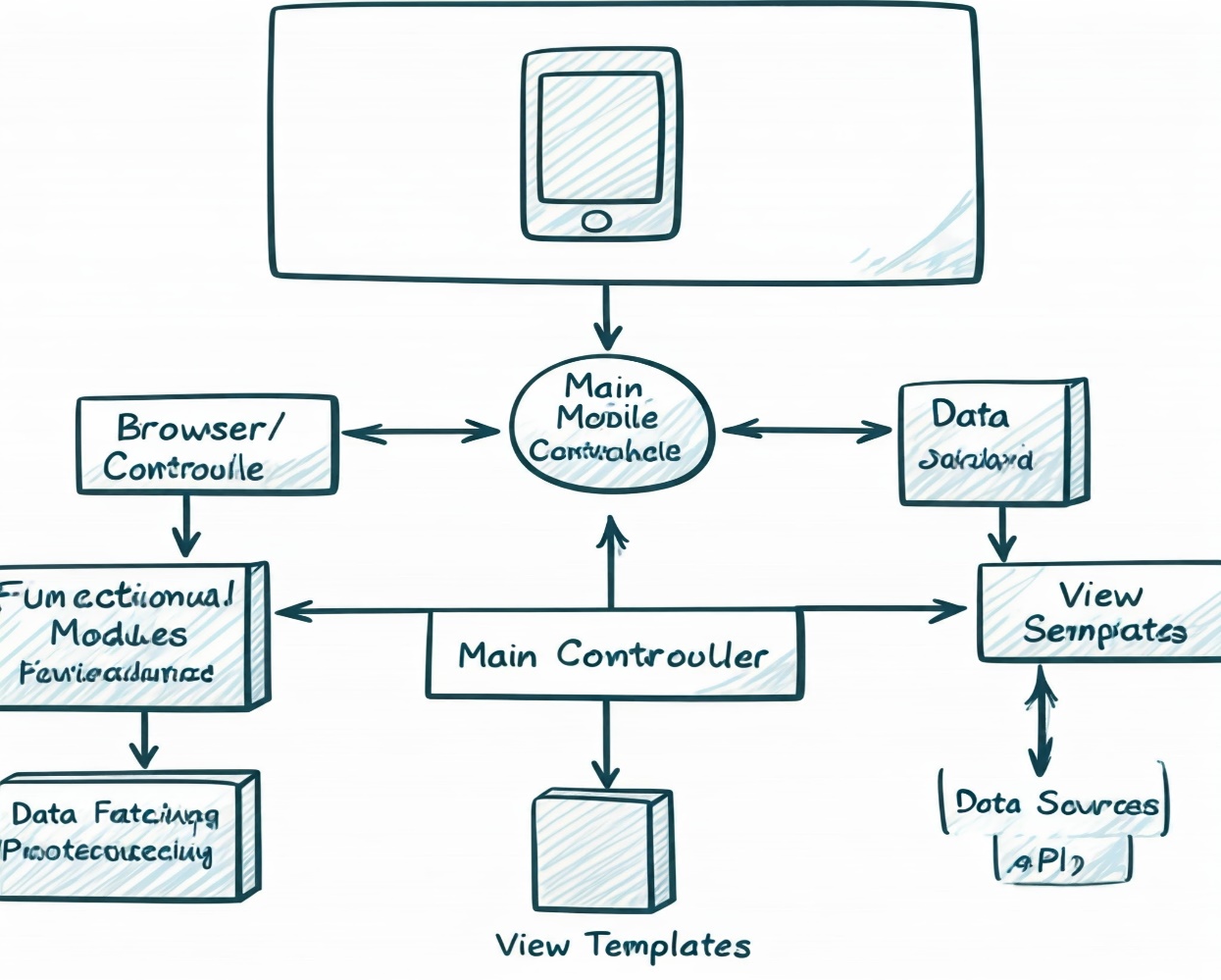
**Figure 3.1 Top-Down Model**

In Figure 3.1, the top-down design approach used in the proposed system is illustrated. This approach begins with the overall structure of the system, where the main components are designed first, followed by a breakdown into smaller, more specific modules. The highest level of the system is responsible for handling core functions, such as user interaction and data management. As the design moves downward, each module handles increasingly detailed tasks.

At the topmost level, the user interacts with the system through a web-based interface, accessible via popular web browsers like Mozilla Firefox, Google Chrome, and Opera. When a user makes a request, such as searching for project information, the system processes the request and either retrieves the necessary data from the database or performs the requested operation. As the request passes through different levels of the system, each module performs its specific role, ensuring a smooth and efficient flow of information.

This top-down approach ensures that all parts of the system work together in a structured manner, where each layer supports the functions of the layers above and below it. Figure 3.2 provides a visual representation of how this process is structured.

**Why this Approach**  
The top-down, modular approach was chosen for its flexibility and simplicity. It allows developers to focus on high-level components before delving into detailed tasks, making the system easier to test, debug, and update. This approach also supports agile development, which speeds up deployment and improves project management.



**Figure 3.2 Step-by-step user request flow the application**

**Figure 3.2** (Workflow of User Request): **Description**: This figure shows the step-by-step process of how users interact with the repository system. The workflow starts with a user making a request (e.g., searching for a project), followed by the system processing the request. Based on the query, the system retrieves relevant data from the database and returns the results to the user.

**CSC Repository Application**

**Student Login**

**Admin Login**

**Project Archive**

**Admin Dashboard**

Search by topic, Student name, Supervisor name or year of submission

Download Project

Add/Edit/View project Entries

Create/Delete Staff Users

Maintenance

**Figure 3.3 Top-down design of the proposed repository application**

**Figure 3.3** (Top-Down Design of the Repository Application): **Description**: This figure illustrates the modular design of the repository application, with different levels representing the various system components. It highlights how user actions, such as logging in or managing projects, flow through the system's layers, ensuring smooth data processing and user interaction.

**3.5 Data Collection**

When the system becomes fully functional, data will be collected from the secretary of the HOD’s office with the approval of the HOD. The goal will be to gather digital copies of past Computer Science Department projects to populate the new database for the repository web application. For demonstration purposes, random data will be used to show how the system works

**Data Security Considerations**

When collecting real project data, strict security measures will be applied to protect sensitive student information. Data encryption and user authentication will be enforced to ensure that only authorized users can access or modify project records..

**3.6 Database Design**

The web application uses MySQL as the database engine. The database's name, structure, and layout are integrated into the application. Table 3.1 shows the user (Student) data table for the proposed system.

**Table 3.1 User (Student) Information table**

|  |  |  |  |
| --- | --- | --- | --- |
| S/NO | FIELD NAME | DATA TYPE | DATA SIZE |
| 1 | Student\_id | Integer | 11 |
| 2 | Student\_name | Varchar | 100 |
| 3 | student\_level | Varchar | 3 |
| 4 | department\_id | Integer | 255 |
| 5 | matric\_number | Integer | 12 |
| 6 | created\_at | Timestamp |  |
| 7 | updated\_at | Timestamp |  |

Table 3.2 is the departments table, which contains information about the departments. This includes the department name, a unique slug, and whether the department should be visible on the application.

**Table 3.2 Departments Table**

|  |  |  |  |
| --- | --- | --- | --- |
| S/NO | FIELD NAME | DATA TYPE | DATA SIZE |
| 1 | department\_id | Integer | 11 |
| 2 | department\_name | Varchar | 50 |
| 3 | visible | Boolean | 1 |
| 4 | created\_at | Timestamp |  |
| 5 | updated\_at | Timestamp |  |

Table 3.3 is the projects table, which contains detailed information about each project. It includes the project topic, project number, a slug (unique identifier), year of submission, student name, and supervisor name, the body (or abstract) of the project, the methodology used, references, and whether the project is visible on the application.

**Table 3.3 Projects table**

|  |  |  |  |
| --- | --- | --- | --- |
| S/NO | FIELD NAME | DATA TYPE | DATA SIZE |
| 1 | project\_id | Integer | 11 |
| 2 | Project\_title | varchar | 255 |
| 3 | year\_of\_submission | Year | 4 |
| 4 | student\_name | Varchar | 100 |
| 5 | surpervisor\_id | varchar | 100 |
| 6 | department\_id | Varchar | 255 |
| 7 | admin\_id | Varchar |  |
| 8 | methodology | Varchar | 255 |
| 9 | visible | BOOLEAN | 1 |
| 10 | created\_at | Timestamp |  |

Table 3.4 is the Admin table it holds information about the administrative users who manage the repository. Each admin is identified by a unique admin\_id, which serves as the primary key. The admin\_name field stores the name of the administrator, while the email field contains the admin’s contact email. Admins are responsible for tasks such as uploading and managing project files. This table helps differentiate between users with administrative privileges and regular users (students).

**Table 3.4 Admin table**

|  |  |  |  |
| --- | --- | --- | --- |
| S/N | FIELD NAME | DATA TYPE | SIZE |
| 1 | admin\_id | Integer | 100 |
| 2 | admin\_name | Varchar | 50 |
| 3 | email | Varchar | 40 |

Table 3.5 is the **Supervisor** table which contains details about the supervisors who oversee student projects. Each supervisor has a unique supervisor\_id, which acts as the primary key. The supervisor\_name field stores the name of the supervisor, and the email field contains their contact information. The department\_id is a foreign key that links each supervisor to a specific department. This relationship ensures that supervisors are correctly associated with their department, which is essential for organizing project supervision within the system

**Table 3.5 Supervisor table**

|  |  |  |  |
| --- | --- | --- | --- |
| S/N | FIELD NAME | DATA TYPE | SIZE |
| 1 | supervisor\_id | Integer | 50 |
| 2 | supervisor\_name | Varchar | 50 |
| 3 | department\_id | Integer | 50 |

### Entity-Relationship (ER) Diagram for the Proposed Repository System

**Why MySQL?**

MySQL is a popular, open-source relational database management system known for its efficiency, scalability, and security. It is well-suited for managing structured data, such as project information, and can easily handle queries that retrieve data based on multiple criteria.

**3.7 Menu Design**

The proposed application will have a streamlined menu for simplicity and functionality. By default, users will be directed to the student login page, with a button available for staff login. The application retain a "Contact" link that routes to a page with relevant contact .The menu will feature:

* **Contact**: Directs users to the contact page with important contact details and an email address.
* **Staff Login**: Positioned at the top-right corner, this link routes to the login page for staff members.

For administrative functions, the admin menu will include:

* **Add a Project**: Allows staff to add new projects to the repository.
* **View All Projects**: Displays all projects in the repository.

**3.8 Query Design**

The proposed repository application allows users to query the database using various parameters. A user can search by a project topic alone, or combine it with the student author’s name, the supervisor’s name, or the year the project was submitted. These queries can be performed individually or combined to produce more refined search results. Figure 3.5 illustrates the different use cases available to users within the system.

**Figure 3.4 Use Case diagram for the proposed system**

Update personal Information

User

Admin

**CHAPTER FOUR**

**SYSTEM IMPLEMENTATION**

**4.1 Choice of Programming Language**

For the development of the proposed repository web application, PHP (Version 8.2.4) and MySQL were selected as the core technologies for the back end. The development environment included Visual Studio Code as the text editor, and XAMPP to handle server and database management. On the front end, HTML, CSS, Tailwind CSS, and Bootstrap were used for designing the user interface.

**Backend Technologies:**

PHP (Version 8.2.4) is a server-side scripting language commonly used for web development. It processes code on a web server using a PHP processor module, generating the necessary output for dynamic web pages. PHP can either be written independently or embedded within HTML code to create dynamic websites. In this project, PHP is utilized to manage server-side logic and database interactions, offering a robust and flexible framework for handling user requests, processing data, and displaying results dynamically.

A top-down design approach was adopted for the system development. This method begins by focusing on the overall structure of the system, then gradually breaks it down into smaller, more manageable components. The top-down approach allows the developer to concentrate on high-level functionality first before diving into more detailed tasks. This ensures a simple, straightforward implementation that is easier to maintain and test.

MySQL serves as the database management system for this project. It is an open-source Relational Database Management System (RDBMS) that relies on Structured Query Language (SQL) for managing and manipulating data. With MySQL, the system is able to perform critical operations such as data insertion, updates, and retrieval efficiently. MySQL's support for both Data Definition Language (DDL) and Data Manipulation Language (DML) ensures that the repository’s data can be properly structured, managed, and queried.

Visual Studio Code was used as the primary text editor for this project due to its versatility and user-friendly interface. It offers features such as syntax highlighting, version control integration, an integrated terminal, and code auto completion, all of which improve development efficiency. Visual Studio Code also allows for extensions and customizations, making it a powerful tool for web development tasks.

**Frontend Technologies:**

The front-end design of the repository web application was developed using HTML, CSS, Tailwind CSS, and Bootstrap.

* HTML (HyperText Markup Language) is the standard markup language used for creating the structure of the web pages. It defines the layout and elements of the web interface, including headers, buttons, and forms.
* CSS (Cascading Style Sheets) is used for styling the HTML elements to ensure a consistent and visually appealing layout. It handles the visual aspects, such as colors, fonts, and spacing.
* Tailwind CSS was the primary CSS framework used in this project. It is a utility-first CSS framework that allows for rapid custom design creation without writing custom CSS for every component. Tailwind CSS made it easier to style elements by providing pre-defined classes for spacing, colors, typography, and more. This resulted in a faster and more responsive design process.
* Bootstrap was used occasionally to complement Tailwind CSS. Bootstrap, a popular CSS framework, was utilized for specific components like responsive grid layouts and pre-designed buttons, ensuring cross-browser compatibility and ease of design implementation.

This combination of HTML, CSS, Tailwind CSS, and Bootstrap enabled the design of a clean, responsive, and user-friendly interface for both students and staff.

XAMPP was used to provide a local development environment that included Apache Server, PHP, and MySQL. XAMPP creates an offline environment where the web application can be developed, tested, and run locally without an internet connection. The "X" in XAMPP stands for cross-platform compatibility, meaning it can be used on various operating systems. The remaining letters in the acronym—"AMPP"—stand for Apache, MySQL, PHP, and other supported languages like Perl or Python.

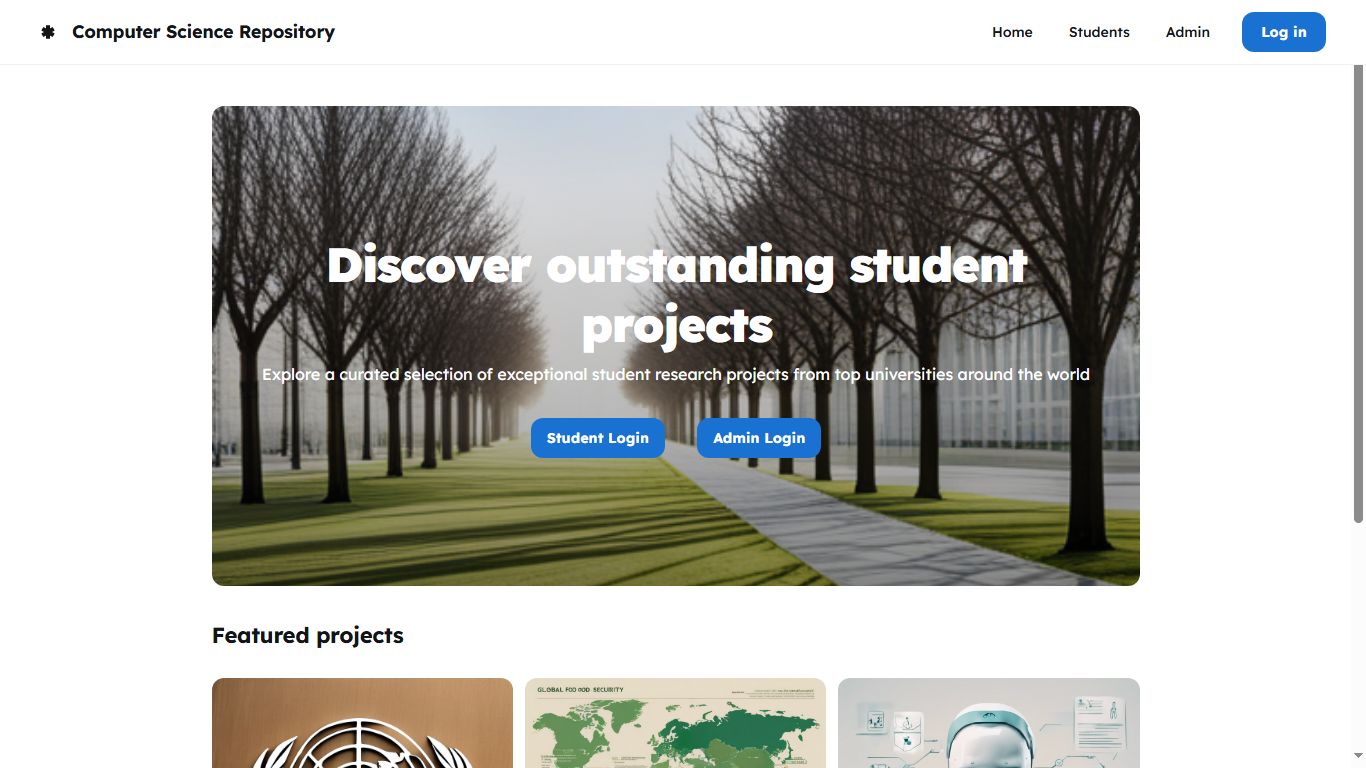
**4.2 The System Main Menu Implementation**

Upon accessing the web application from a browser, the landing page is presented. The landing page features a top main navigation menu, an eye-catching image, and two prominent buttons for Student Login and Admin Login. The landing page introduces the repository with a welcoming note that reads: "Discover outstanding student projects," highlighting the repository's curated selection of student research projects from top universities.

The Computer Science Repository is structured to serve both students and staff in a modular form. The top-right corner of the landing page features a “Log in” button, which provides access to both the Student Login and Admin Login sections.

* Student Login: Students access the repository by logging in with their matriculation number and a password. Once authenticated, they are directed to their respective dashboard, where they can view, search, and download available project materials.
* Admin Login: the administrator logs in with a unique code and a password. The single admin has full access to all features of the repository, including managing student and project data. Admin privileges include viewing, creating, and deleting project entries, managing student access, and handling system-wide settings.

This modular design ensures that the repository remains flexible and secure, allowing easy access to different functionalities based on the role of the user (admin or student).



**Figure 4.1 Landing Page**

**4.3 Staff Login Implementation**

The Admin Login page, as shown below, is a simple and clean interface requiring two essential inputs: the Admin Number and Password. Upon visiting this page, staff members are prompted to enter their credentials in the provided fields.

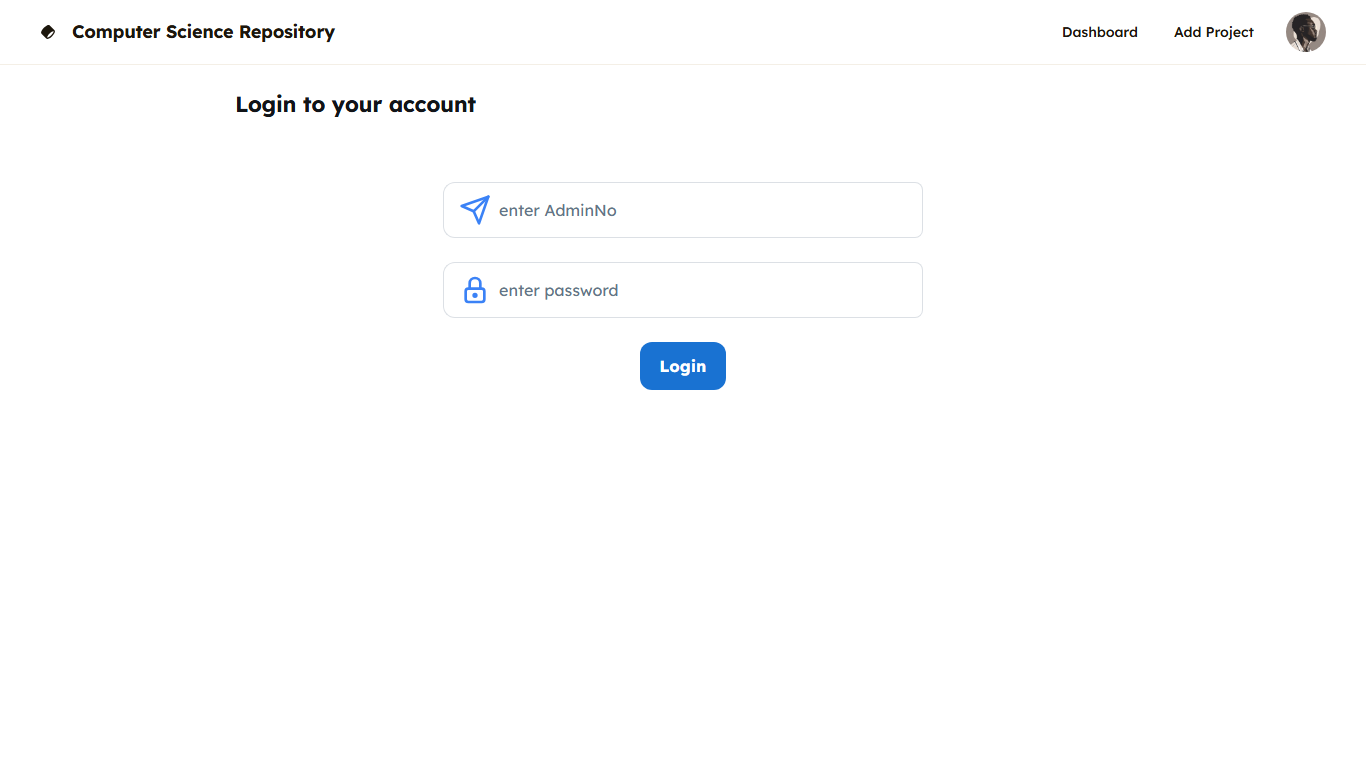
**1. Security Features**: The login system implements basic security protocols to protect user credentials. Every login request is validated, and measures are in place to prevent unauthorized access. For instance:

* **Credential** **Validation**: The system checks that the Admin Number and Password match valid entries in the database before granting access. Incorrect login attempts return appropriate error messages, ensuring the user is aware of invalid inputs.
* **Password** **Encryption**: Passwords are encrypted to ensure they are securely stored in the system. On login, the system compares the encrypted input with the stored password to confirm authenticity.

**2.** **Login Process**: When a user provides their Admin Number and Password, the system cross-checks the credentials with the database. On successful verification, the user is redirected to their Dashboard which provides access to the admin functionalities. Invalid credentials result in an error message prompting the user to try again.

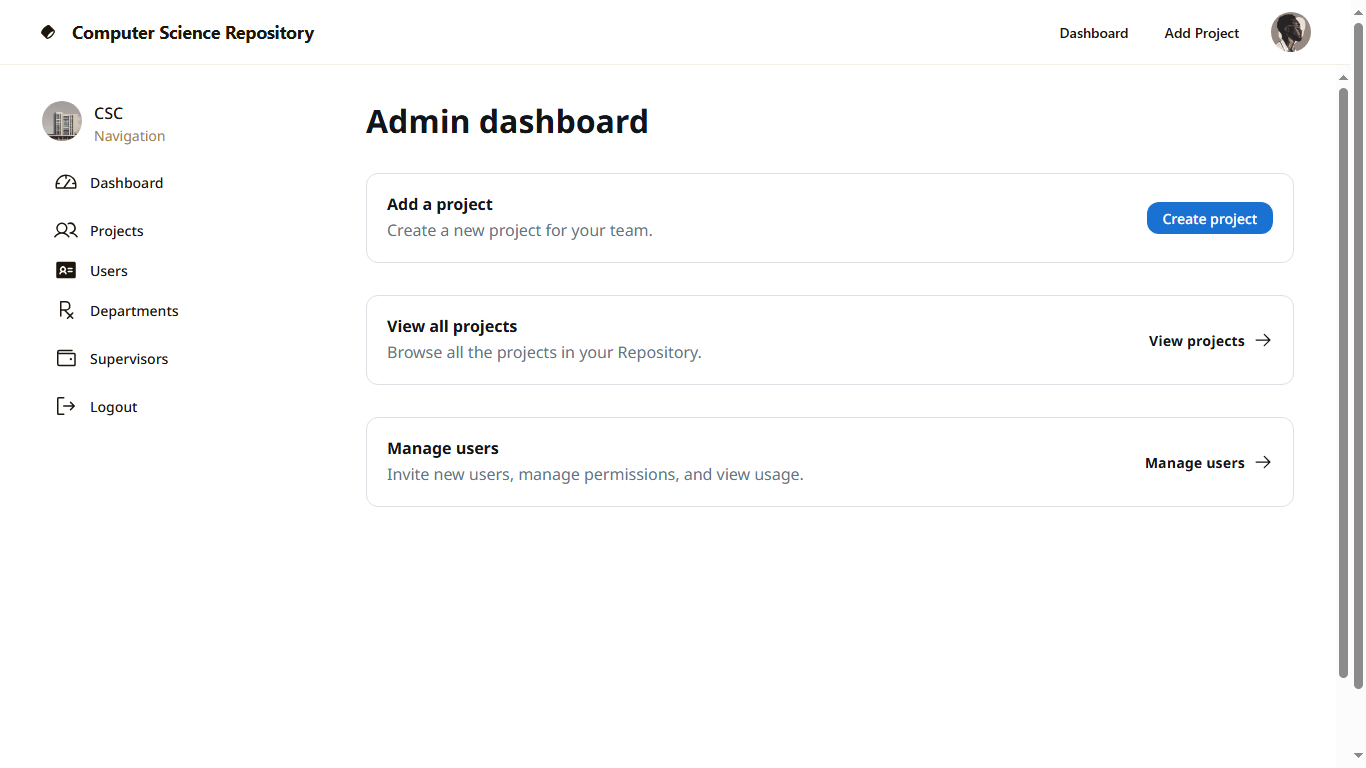
**3. User** **Feedback**: The page offers user feedback in the form of clear error messages in the event of incorrect credentials, ensuring a smooth user experience. Additionally, the system handles errors gracefully, guiding the user on how to proceed in case of any issues.

**4. User Redirection**: Upon successful login, the staff member is redirected to the Dashboard. Here, staff members can perform administrative tasks such as uploading new projects, managing existing ones, and overseeing user activities.



**Figure 4.2 Staff Login Page**

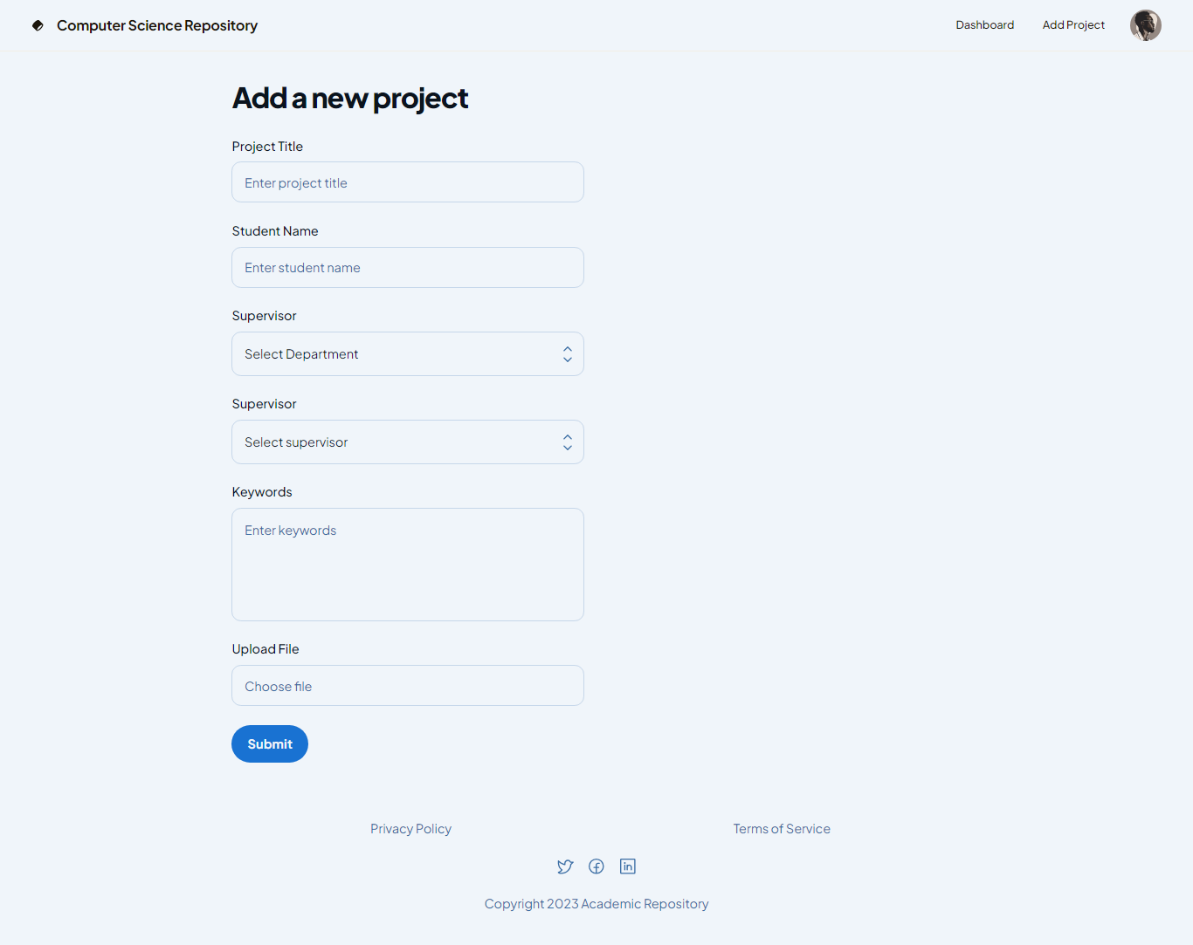
**4.4 Implementation of the Sub-System**

Clicking the "Admin Login" button on the landing page redirects the user to the Staff Login interface. Upon entering valid credentials (Admin Number and Password), access is granted, leading to the dashboard. The dashboard contains a vertically situated menu on the left side of the page, from which the staff can add a project, view all projects, or update a project. The layout provides seamless navigation and efficient access to core functionalities of the repository. 

**Figure 4.3 Staff Dashboard Interface**

**4.5 Project Creation**

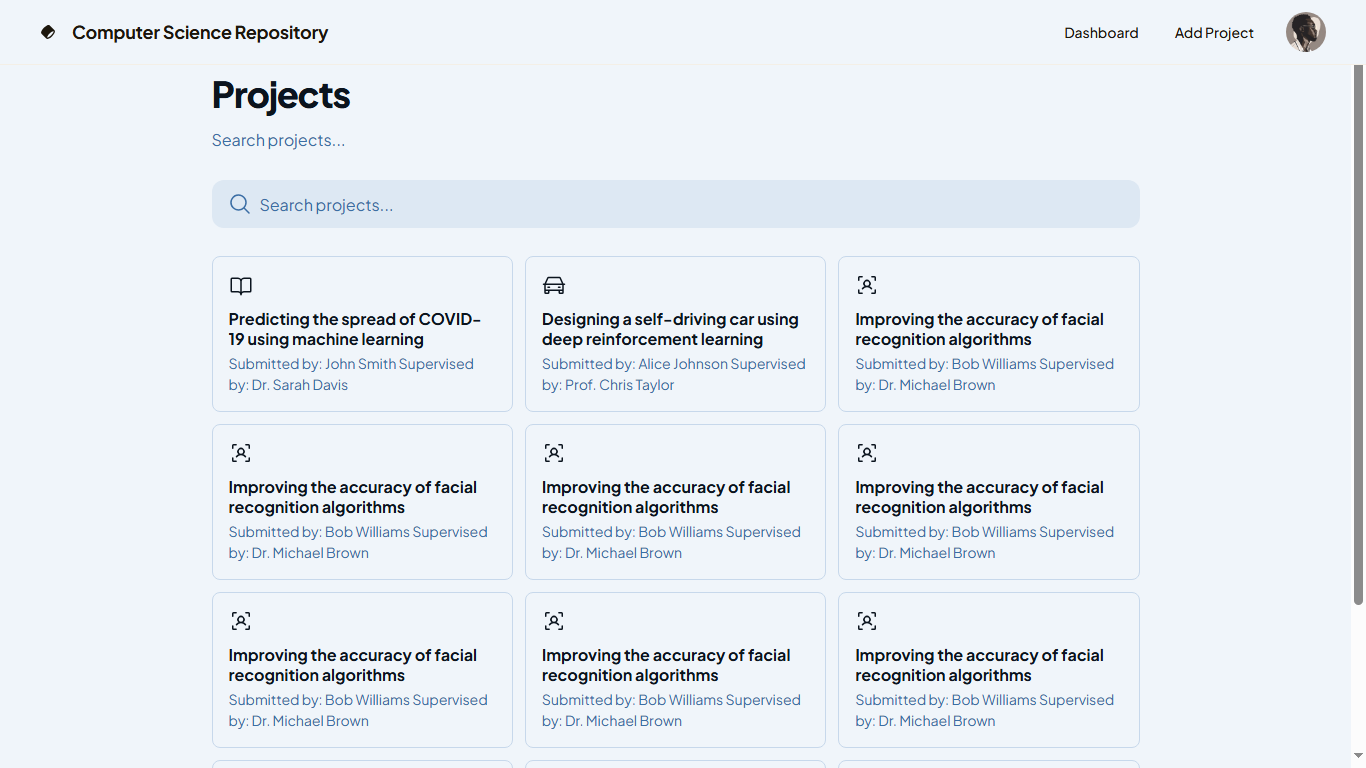
Adding a project to the repository is a straightforward process. The staff user is required to input relevant project metadata, such as the project title, author, supervisor, introduction, and other necessary fields, into the provided input boxes. Once the information is entered, clicking the "Submit" button submits the data, creating a new entry in the database that stores the project details. The simplicity of the system ensures that a staff user, even with basic computer skills, can easily navigate the project creation process. The system provides user-friendly input fields and clear instructions, making the process efficient after just a few attempts. Error handling is implemented to manage required fields and prevent invalid data entry, with readable error messages displayed in simple language. Upon successfully creating a project, a success message is shown at the top of the page, confirming that the project has been saved into the repository.



**Figure 4.4 Project Creation Form**

**4.6 Query Sub-System Implementation**

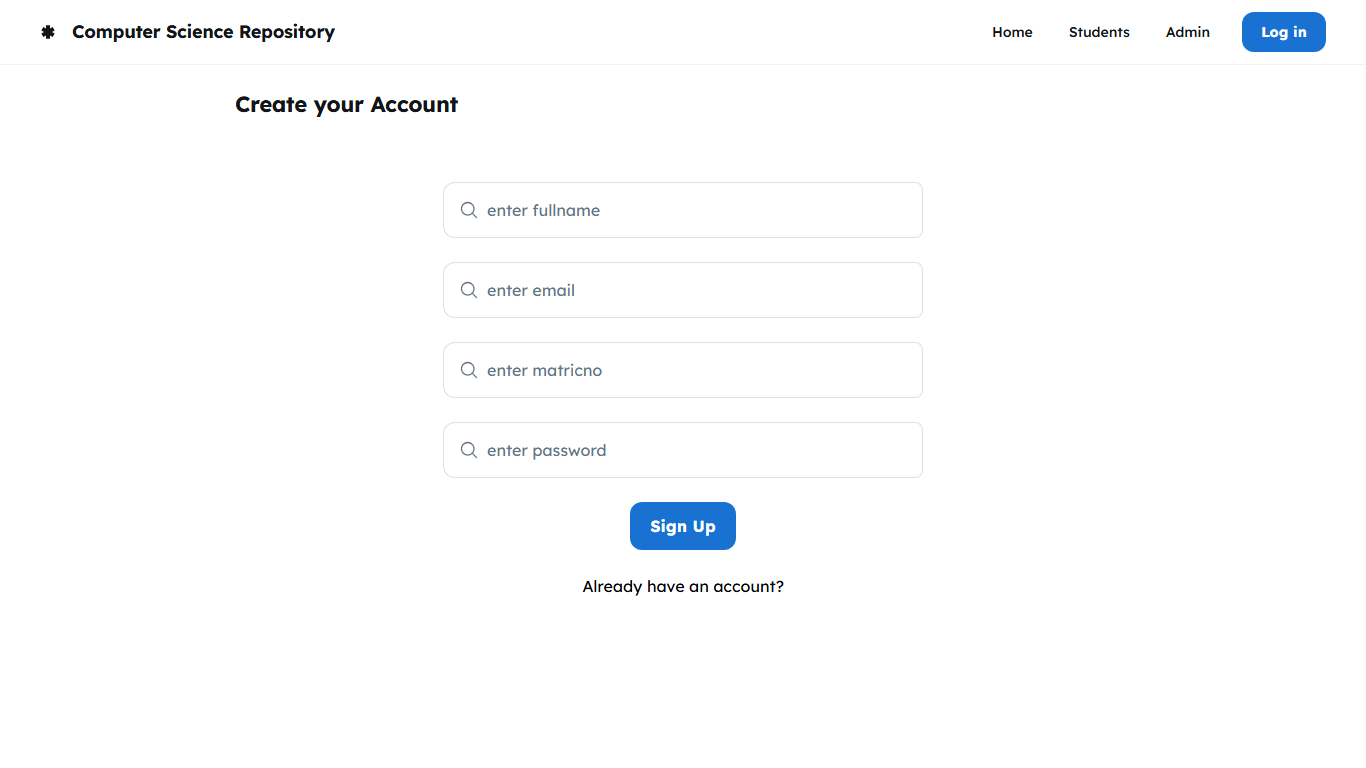
Once projects are added to the repository, users can easily query projects by title, year, supervisor, and student. The query system is designed to be flexible, accommodating various search criteria to quickly retrieve specific project information.



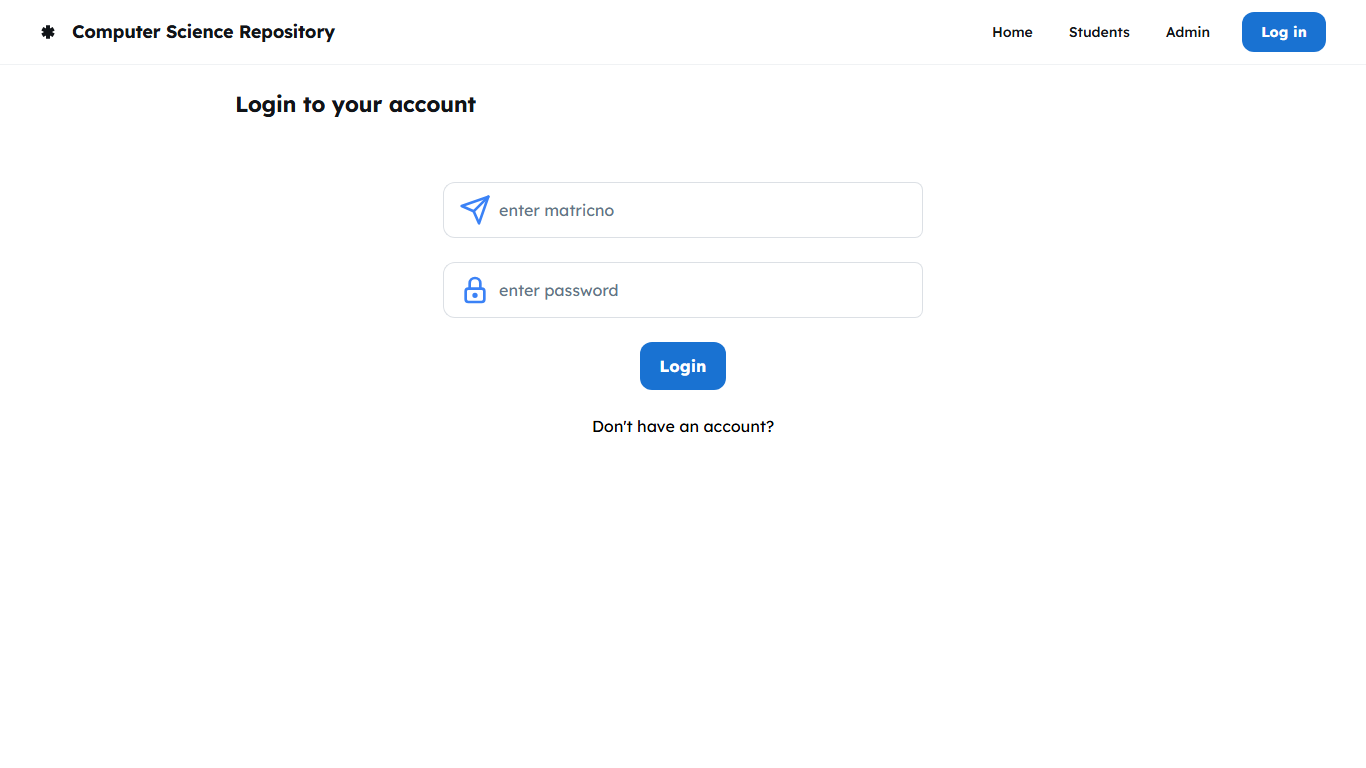
**Figure 4.5 Project Search/Dashboard**

**Student Section**

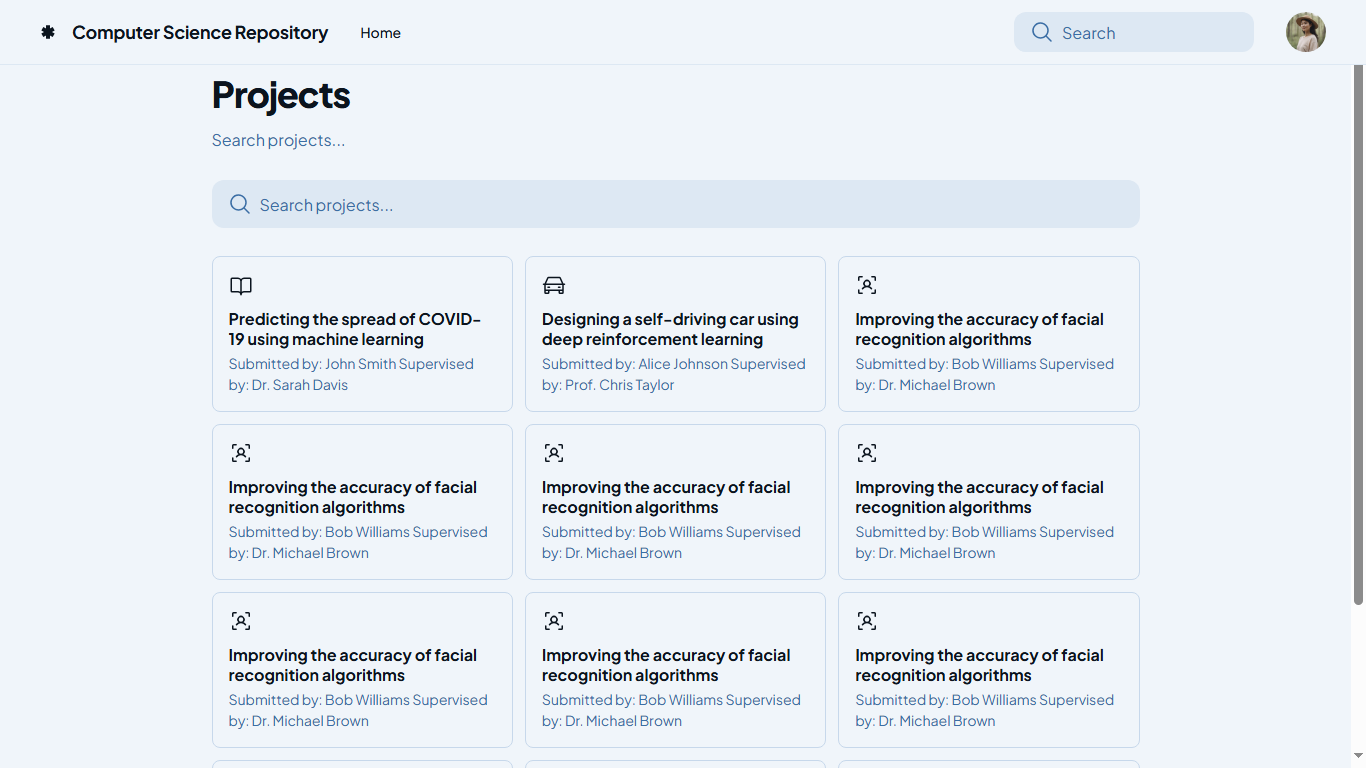
The student section of the repository allows students to log in using their matriculation number and password. Once authenticated, students can browse and download past projects based on criteria such as department or year. The interface is user-friendly and intuitive, making it easy for students to quickly access and download project files. Security is ensured through proper authentication measures, and students can only view projects after successful login.



**Figure 4.6 Student sign-up page**



**Figure 4.7 Student login page**

**  
Figure 4.8 Student Dashboard**

**4.7 System Evaluation**

The current system with the proposed (new) system were compared to show a clear and easy to note contrast of the two the two systems. Table 4.1 shows evaluations of the current system against the new system, from the results obtained, it can be said that the new system is a true improvement on the current system and gives more efficiency in storing and retrieving of undergraduate projects. The values used to evaluate the current system were obtained by intuition, therefore all calculations of the current system are close estimates.

**Table 4.1 System comparison**

|  |  |  |
| --- | --- | --- |
| **Evaluated under** | **Current System** | **New System** |
| 1. Time taken to search for one undergraduate project (using 50 projects to evaluate) | Assuming checking one project’s details take 5 seconds approximately.  Time Taken:  5 \* 50 = 250 seconds | Time taken to search one project is 0.0023 seconds.  Time taken:  0.0023 \* 50 = 0.115 seconds |
| 1. Water and Fire disaster | Some or all of the archived undergraduate projects will be damaged or totally lost. | Water and Fire cannot affect this system because it can be hosted online. |
| 1. Official Procedures | A formal is needed to be addressed to the Head of Department of a department before access is granted | No official procedure is needed, only a web accessing device with an active internet connection. |
| 1. Time taken to archive one undergraduate project (using 50 projects to evaluate) | Assuming it takes 3 seconds to set one physical copy of a project on the archive shelf. Time taken:  3 \* 50 = 150 seconds. | Time taken to add/create one project is 0.534 seconds.  Time taken:  0.534 \* 50 = 26.7 seconds. |
| 1. Time bound | Cannot be accessed anytime, complete undergraduate projects works given out have a time bound for return. | Can be accessed anytime when online. There is no time bound to when to return metadata for an undergraduate project work. |

References

Abrizah, A. (2009). The cautious faculty: their awareness and attitudes towards institutional repositories. Malaysian Journal of Library & Information Science, Vol. 14, no. 2, 17-37.

Bull, J., & Eden, B. L. (2014). Successful scholarly communication at a small university: Integration of education, services, and an institutional repository at Valparaiso University. College & Undergraduate Libraries, 21(3-4), 263-278.

Digital Repository Software (2012). Eprints – Digital Repository Software. Retrieved October 15, 2015 from http://www.eprints.org/software/

Dora, M. & Maharana, B. (2012), Driving on the Green Road: Self-archiving Research for Open Access in India (2012). Library Philosophy and Practice (e-journal).Paper 785. Retrieved from http://digitalcommons.unl.edu/libphilprac/785

DSpace (2008, December 28). Wikipedia, the free encyclopedia. Retrieved July 20, 2015 from https://en.wikipedia.org/w/index.php?title=DSpace&oldid=260510011

English, E. (2010), A Digital Repository at Loyola University Chicago (2010). Digital Services Librarian, 1-10.

EPrints (2015, March 15). Wikipedia, the free encyclopedia. Retrieved July 20, 2015 from https://en.wikipedia.org/wiki/EPrints

Gumpenberger, C. (2009). The eprints story: Southampton as the cradle of institutional self-archiving. GMS Med Bibl Inf, 9(1). Retrieved April 23, 2015 from http://eprints.ecs.soton.ac.uk/17581/1/The\_EPrints\_story\_Gumpenberger\_final\_draft.doc

Institutional Repository (2014, April 16). New World Encyclopedia: Research begins here. Retrieved July 15, 2015 from http://www.newworldencyclopedia.org/entry/Institutional\_repository

Mukhlesur Rahman, M., & Mezbah-ul-Islam, M. (2014). Issues and strategy of institutional repositories (IR) in Bangladesh: a paradigm shift. The Electronic Library, 32(1), 47-61.

Open Access Repositories - Open Access Scholarly Information Sourcebook (n.d.). Retrieved July 12, 2015 from http://www.openoasis.org/index.php?option=com\_content&view=article&id=137&Itemid=353

Peake, M. (2012), Open Archives Initiative Protocol for Metadata Harvesting, Dublin Core and Accessibility in the OAIster Repository. Library Philosophy and Practice (e-journal). Paper 892. Retrieved from http://digitalcommons.unl.edu/libphilprac/892

Richards, G., McGreal, R. & Friesen, N. (2002), Learning Object Repository Technologies for TeleLearning: The Evolution of POOL and CanCore. Informing Science, 1333-1338.

Swan, A. (2009), Open Access institutional repositories: A Brieﬁng Paper. Retrieved from http://www.openscholarship.org/upload/docs/application/pdf/2009-01/open\_access\_institutional\_repositories.pdf

Tansley, R., Smith, M. & Harford, J. W. (2005), The DSpace Open Source Digital Asset Management System: Challenges and Opportunities. Hewlett-Packard Laboratories, Cambridge Center, Cambridge, MA 02142, MIT Libraries, 77 Massachusetts Ave, Cambridge, MA 02139.

Thorat, S. V., & Patil, S. K. (2011). Institutional Repository: A Proposed Model for Bharati Vidyapeeth Deemed University, Pune. Retrieved July 20, 2015 from http://ir.inflibnet.ac.in:8080/dxml/bitstream/handle/1944/1625/34.pdf?sequence=1

Williams, L., Pope, K. & Lucero, B. L. (2014), Institutional repositories provide an ideal medium for scholars to move beyond the journal article. Academic communication, Open Access. Retrieved from http://blogs.lse.ac.uk/impactofsocialsciences/2014/03/12/institutional-repositories-move-beyond-the-journal-article/