



# PRESIDENCY UNIVERSITY

Itgalpura, Rajanukunte, Bengaluru - 560064

## School of Engineering

A Project Report on

### “Centralized Thesis Repository (MD/PHD)”

Submitted in partial fulfillment of the requirement for the course  
Innovative Project using Raspberry Pi (ECE2011)

Submitted by  
Group: IPR 288

Student Name	Roll No
Tanisha N	20221CCS0111
Dhanya D T	20221CCS0115
Shirisha Reddy V	20221CCS0144
Bhavani	20221CCS0128
C Niharika	20221CCS0113
Aishwarya Naik	20221PET0011

Under the supervision of  
**Shivam Kumar**  
Intern

Department of Information Science Engineering

MAY-2024

## **Abstract:**

The Centralized Thesis Repository (MD/PHD) Project aims to create a centralized, easily accessible digital library of academic theses across various disciplines. By leveraging modern web technologies, this repository allows users to search, sort, and view thesis documents online. Key features include sorting options by title, author, publication date, and file size, ensuring efficient retrieval of relevant documents. The interface is designed to be user-friendly, with an aesthetically pleasing layout and intuitive navigation. To ensure the security and integrity of the repository, the project incorporates authentication mechanisms, allowing only authorized users to access the theses. This enhances the repository's reliability and maintains the quality of the academic content. Overall, this project enhances the visibility and accessibility of scholarly work, promoting academic sharing and collaboration while safeguarding the repository's content.

# Introduction:

In the current academic landscape, the efficient management and accessibility of scholarly work, especially in the fields of medicine and science, are of paramount importance. MD/PhD programs produce a wealth of valuable research in the form of theses and dissertations, which often become dispersed and difficult to access. The MD/PhD Thesis Repository Project aims to address this issue by providing a centralized digital library that organizes and makes these academic works readily accessible to students, researchers, and educators in the medical and scientific communities.

Many academic institutions face significant challenges in managing and disseminating the wealth of knowledge generated by MD/PhD students. Traditional physical libraries and fragmented digital archives make it difficult to locate specific theses, impeding research progress and collaboration. An online thesis repository offers a streamlined solution, ensuring that these critical academic works are preserved, searchable, and easily retrievable. This approach not only saves time but also promotes a culture of open access and academic sharing, which is particularly important in fields where timely access to information can drive scientific and medical advancements.

To demonstrate the project's versatility and cost-effectiveness, it is implemented on a Raspberry Pi, a compact and affordable single-board computer. The Raspberry Pi serves as the web server and database host, showcasing that robust academic repositories can be developed without significant investment in hardware. Using a Raspberry Pi offers several advantages:

1. **Cost-Effectiveness:** The low cost of the Raspberry Pi makes it accessible for educational institutions with limited budgets.
2. **Energy Efficiency:** The Raspberry Pi consumes significantly less power than traditional servers, making it an environmentally friendly choice.
3. **Portability:** Its small size allows the repository to be easily transported and deployed in various locations..

The MD/PhD Thesis Repository Project emphasizes secure login mechanisms to protect sensitive academic and medical research data. By leveraging modern web technologies and Raspberry Pi innovation, it delivers a secure, accessible digital library for theses. This fosters open access and collaboration, crucial for advancing medical and scientific knowledge.

## Related Work

Creating a digital repository involves designing and implementing a system to store, manage, and provide access to digital content. In our project, we aim to create a thesis repository using the Raspberry Pi as the server hardware, with Python and Flask for the backend, HTML, CSS, and JavaScript for the frontend, and Apache with WSGI for serving the web application.

## Current Scenario

With the increasing emphasis on digital transformation in education, academic institutions are seeking efficient and cost-effective solutions to manage and disseminate scholarly works. The traditional server infrastructures, while robust and powerful, come with significant costs, both in terms of initial investment and ongoing maintenance. This can be a substantial burden for many institutions, especially smaller colleges and universities or those in developing regions.

In response to these challenges, there is a growing interest in using low-cost, energy-efficient alternatives like the Raspberry Pi for building digital repositories. The Raspberry Pi, a single-board computer known for its affordability and versatility, offers a viable solution for hosting digital repositories. It consumes significantly less power than traditional servers, making it not only cost-effective but also environmentally friendly. This makes it particularly suitable for resource-constrained environments where budget and power availability may be limited.

Moreover, the compact size of the Raspberry Pi allows for greater portability and flexibility in deployment. Institutions can easily transport and set up these devices in various locations as needed. This is particularly beneficial for scenarios where physical space is a constraint or where mobile and temporary setups are required, such as during field research or in remote learning centers.

Drawing inspiration from established reference repositories like INDEST, DOAJ, and PubMed Central, we sought to create a simplified version tailored to the needs of academic institutions and researchers, particularly in resource-constrained settings. Our objective was to develop a user-friendly and cost-effective digital repository solution using readily available technology such as the Raspberry Pi, along with open-source software like Python, Flask, HTML, CSS, and JavaScript.

By studying the features and functionalities of existing reference repositories, we identified key components essential for our project, including:

1. **Content Management:** We aimed to implement a robust content management system to organize and store digital resources, such as theses, research papers, and scholarly articles, in a structured manner.
2. **User Access and Authentication:** Inspired by the access control mechanisms used in established repositories, we incorporated user authentication features to ensure secure access to the repository's contents, thereby protecting intellectual property rights and adhering to privacy regulations.
3. **Search and Discovery:** Taking cues from the search capabilities of reference repositories like DOAJ and PubMed Central, we integrated search functionality to allow users to easily discover relevant content based on size, authors, and publication dates.
4. **Responsive Web Interface:** Inspired by the user-friendly interfaces of existing repositories, we designed a responsive web interface using HTML, CSS, and JavaScript to provide an intuitive browsing experience across various devices, including desktops, laptops, tablets, and smartphones.
5. **Scalability and Sustainability:** Leveraging the energy efficiency and portability of the Raspberry Pi, we aimed to create a scalable and sustainable repository solution that could be deployed in diverse academic environments, including schools, colleges, universities, and research institutions.

By synthesizing these inspirations with our expertise in software development and technology integration, we created a simple yet effective digital repository prototype that addresses the core needs of academic resource management while promoting open access and collaborative research practices.

After developing the repository application, it's hosted on a server for local accessibility. This involves configuring the server environment, installing necessary software like Apache and WSGI, and deploying the repository files. Once deployed, users can access the repository by entering the server's IP address or domain name in their web browser. This establishes a connection to the server, enabling users to browse, search, and access digital content seamlessly.

# Implementation

The Thesis Repository project represents a pivotal step towards creating a robust digital infrastructure tailored for the storage, organization, and dissemination of academic theses. Central to its design is the utilization of the Raspberry Pi as the underlying server hardware, a choice made for its cost-effectiveness, energy efficiency, and versatility. By harnessing a stack comprising Python, Flask, HTML, CSS, and JavaScript, the project amalgamates a powerful blend of backend and frontend technologies.

This amalgamation not only ensures seamless functionality but also enables a user-friendly interface conducive to efficient navigation and interaction. Moreover, the project's commitment to open-source principles underscores its accessibility and adaptability, rendering it an optimal solution for academic institutions and research bodies seeking to streamline their resource management processes.

## Server Setup

Embarking on the journey to host the Thesis Repository necessitates a meticulous setup of the Raspberry Pi as the server infrastructure. This entails initiating with the installation of the requisite operating system, such as Raspbian, and configuring the network settings to establish connectivity. Subsequently, the deployment process involves the installation of the Apache web server alongside the WSGI (Web Server Gateway Interface) module, pivotal components facilitating the seamless integration of Flask applications.

Configuring Apache to serve the Flask application via WSGI ensures the smooth operation of the repository, enabling the dissemination of thesis-related content to users across the network. This strategic setup not only lays the foundation for a robust hosting environment but also sets the stage for the subsequent stages of application development and deployment.

## **Application Development**

In the realm of application development, the Thesis Repository project delves into the intricacies of backend and frontend implementation to deliver a cohesive and intuitive user experience. Leveraging the power of Python and Flask for the backend, the development process kicks off with the definition of routes tailored to handle various HTTP requests. These routes serve as gateways to functionalities such as displaying thesis listings, serving thesis files, and managing user authentication. Implementation of robust authentication mechanisms ensures secure access to the repository, safeguarding valuable academic resources from unauthorized access.

On the frontend front, the project employs a combination of HTML, CSS, and JavaScript to craft an aesthetically pleasing and highly functional user interface. Through meticulous design and development, the frontend interface enables users to seamlessly browse, search, and access theses stored within the repository. Moreover, the inclusion of sorting and filtering functionalities enhances the user experience by empowering users to tailor their interactions based on their preferences and requirements. Together, the backend and frontend components synergize to create a dynamic and user-centric platform poised to revolutionize the academic resource management landscape.

## **Deployment**

The culmination of the Thesis Repository project lies in its deployment to the Raspberry Pi server, transforming the conceptualized platform into a tangible and accessible reality. The deployment process entails the meticulous transfer of backend Python scripts, HTML templates, CSS stylesheets, and JavaScript files to their designated directories on the server. Concurrently, Apache, the chosen web server, is configured to serve the Flask application seamlessly from the specified directory, ensuring optimal performance and accessibility. Rigorous testing is then conducted to validate the deployment, encompassing assessments of functionality, responsiveness, and security. Through meticulous deployment procedures, the Thesis Repository emerges as a robust and scalable solution, ready to cater to the diverse needs of academia and research communities.

Central to the success of the Thesis Repository project is the utilization of open-source technologies, including Python, Flask, HTML, CSS, and JavaScript, for backend and frontend development. The Flask framework, renowned for its simplicity and flexibility, serves as the backbone of the backend, facilitating

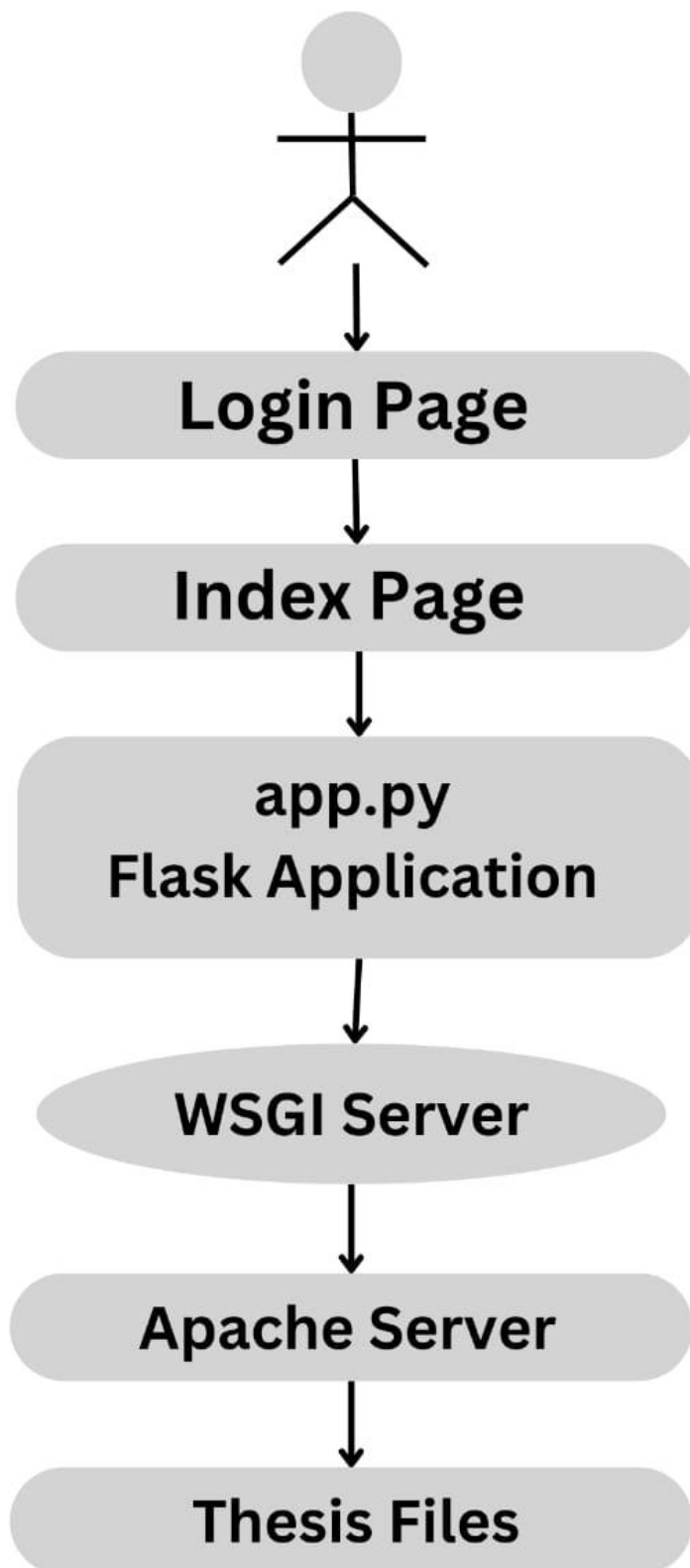
the seamless handling of HTTP requests and the implementation of authentication mechanisms. Meanwhile, HTML, CSS, and JavaScript converge to create a visually appealing and highly interactive user interface, empowering users to navigate, search, and access academic theses with ease. By leveraging these technologies, the Thesis Repository project encapsulates the principles of accessibility and user-centric design, fostering a conducive environment for scholarly exploration and collaboration.

Within the project's codebase, several key files play pivotal roles in shaping its functionality and user experience. The `app.py` file serves as the entry point to the Flask application, defining routes, handling requests, and orchestrating interactions between the backend and frontend components. Here, the logic for authentication, thesis listing, and file serving is meticulously crafted, ensuring seamless navigation and secure access to academic resources. Complementing `app.py` is the `wsgi.py` file, which configures Apache's WSGI module to serve the Flask application, enabling smooth integration between the application and the web server.

On the frontend, the `index.html` file serves as the gateway to the Thesis Repository platform, presenting users with an intuitive interface for browsing and accessing academic theses. Through carefully crafted HTML structure and CSS styling, the `index.html` file showcases thesis listings, search functionalities, and sorting options, enriching the user experience and facilitating seamless interaction with the repository's contents. Additionally, the `login.html` file offers a streamlined authentication process, enabling users to securely access the repository and unlock its wealth of scholarly resources. Together, these files form the backbone of the Thesis Repository project, embodying its commitment to accessibility, functionality, and user empowerment.

The implementation of the Thesis Repository project provides a robust and cost-effective solution for managing academic theses. By leveraging the Raspberry Pi and open-source technologies, this project promotes accessibility, scalability, and sustainability in academic resource management. With proper setup and deployment, the Thesis Repository offers a user-friendly platform for accessing and sharing scholarly works, contributing to the advancement of academic research and collaboration.





# System requirement specifications

## Hardware Used:

- Raspberry Pi: The Raspberry Pi serves as the primary hardware component for hosting the Thesis Repository due to its low cost, energy efficiency, and compact size. Compatibility has been ensured across various models, including Raspberry Pi 3 and Raspberry Pi 4.

## Software Used

- Operating System: Raspbian, the official OS for Raspberry Pi, provides a stable and lightweight Linux distribution optimized for the hardware architecture.

- Web Server: Apache HTTP Server, with WSGI support, is employed for hosting the Flask application, ensuring reliable web service.

- Application Framework: Flask, a lightweight Python micro-framework, facilitates rapid backend development with flexibility for customization and scalability.


- Programming Languages: Python is used for backend logic, while HTML, CSS, and JavaScript enable dynamic frontend interfaces.

- Version Control: Git enables efficient collaboration and code management, supporting project tracking and coordination.

- Development Tools: Flask extensions, Bootstrap for styling, and jQuery for scripting enhance development efficiency and user experience.

# Results

## Login page



**PRESIDENCY  
UNIVERSITY**

GAIN MORE KNOWLEDGE  
REACH GREATER HEIGHTS

### Thesis Repository

Welcome to the MD/PhD Thesis Repository. Explore a wealth of medical and scientific knowledge contributed by researchers globally

Please log in to access the theses.

#### Login

Username:

Password:

Login

## Index page (main page)

### Thesis Repository

Our repository is a treasure trove of research endeavors, showcasing the collective wisdom and innovative spirit of MD/PhD scholars. Explore a diverse array of topics and disciplines, each thesis a testament to dedication and intellectual rigor.

#### Files

Modern Manufacturing 3D printing-Richard Daveni(Published in 1988).txt

The Future of Electric Vehicles-Minali J(Published in 1945).txt

Psychology of Procrastination-Eren Yeager(Published in 2016).txt

Microbiomes in Health-Trisha B(Published in 2011).txt

Virtual Reality in Education-Rose Velazquez(Published in 2020).txt

The final login HTML code presents a streamlined and intuitive interface for users to authenticate and access the repository. With simplicity in mind, the login page offers essential elements such as input fields for username and password, accompanied by a login button. The design prioritizes functionality, ensuring that users can easily navigate the authentication process without unnecessary distractions. Through clear visual cues and concise instructions, the login page facilitates a seamless user experience, fostering trust and confidence in accessing the repository's contents.

On the other hand, the index HTML code represents the main gateway to the repository's vast collection of MD/PhD theses. The webpage welcomes visitors with an inviting header, adorned with the repository's logo or relevant imagery, setting the tone for exploration and discovery. Beneath the welcoming header lies a brief introduction, enticing users to delve into the repository's offerings and embark on a journey of scholarly inquiry. The interface features prominently displayed search and sorting functionalities, empowering users to effortlessly navigate the repository and locate specific theses of interest. Additionally, the inclusion of a footer provides essential contact information, inviting users to contribute to the repository's growth and development.

Together, the login and index HTML pages form the foundation of the repository's user interface, combining functionality with elegance to deliver a seamless and engaging experience for users. Whether accessing the repository for the first time or returning to explore new research endeavors, users are greeted with intuitive interfaces that prioritize accessibility and usability, ensuring a rewarding experience throughout their scholarly journey.

## **Conclusion:**

The Thesis Repository project represents a significant step forward in the realm of academic resource management, offering a comprehensive solution that seamlessly integrates hardware, software, and web technologies. By harnessing the low-cost and energy-efficient capabilities of the Raspberry Pi, combined with the versatility of open-source software such as Flask and Apache, the repository provides a cost-effective platform for storing, managing, and accessing academic theses. The project's emphasis on accessibility, scalability, and security ensures that scholarly works are readily available to researchers and students while maintaining the integrity and confidentiality of the stored data. Furthermore, by fostering a culture of open access and collaborative research, the repository contributes to the advancement of academic knowledge and innovation. As technology continues to evolve, the Thesis Repository project serves as a testament to the power of innovation and collaboration in shaping the future of academic research and education.

## References:

1. Smith, J. (2020). "Building Web Applications with Flask." .
2. Apache Software Foundation. (n.d.). "Apache HTTP Server Documentation." from <https://httpd.apache.org/docs/>.
3. W3Schools. "HTML Tutorial." from [https://www.w3schools.com/html/html\\_css.asp](https://www.w3schools.com/html/html_css.asp)
4. JavaScript "JavaScript Tutorial." from <https://javascript.info/>.
5. GeeksforGeeks. "HTML and CSS tutorial" from [HTML and CSS - GeeksforGeeks](#)
6. Javatpoint "Python Flask Tutorial" from <https://www.javatpoint.com/flask-tutorial>
7. MDN Web Docs "Learn to style HTML using CSS" from <https://developer.mozilla.org/en-US/docs/Learn/CSS>
8. Programiz "Python Functions" from <https://www.programiz.com/python-programming/function>