A

Mini Project Report

on

APSIT AR NAVIGATION SYSTEM

Submitted in partial fulfillment of the requirements for the degree

of

Third Year Engineering – Information Technology

by

Dhruv Mahyavanshi	**22104040**
Kartik Patil	**22104023**
Harsh Prajapati	**22104188**

Under the guidance of

Prof. Jayshree Jha

DEPARTMENT OF INFORMATION TECHNOLOGY

A.P. SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W)-400615

UNIVERSITY OF MUMBAI

Academic Year: 2024-25

CERTIFICATE

This to certify that the Mini Project report on APSIT AR NAVIGATION SYSTEM has been submitted by Kartik Patil (22104023), Dhruv Mahyavanshi (22104040) and Harsh Prajapati (22104188) who are bonafide students of A. P. Shah Institute of Technology, Thane as a partial fulfillment of the requirement for the degree in Information Technology, during the academic year 2024-2025 in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Prof. Jayshree Jha
Guide

Dr. Kiran Deshpande
HOD, Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner:
1.

Internal Examiner:
1.

Place: A. P. Shah Institute of Technology, Thane
Date:

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide **Prof. Jayshree Jha**. Expressing gratitude towards our HoD, **Dr. Kiran Deshpande**, and the Department of Information Technology for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our project coordinators **Ms. Sonal Jain & Ms. Shafaque Syed** who gave us his valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

TABLE OF CONTENTS

ABSTRACT

The AR Navigation System with Chatbot and Learning Platform is a web-based application designed to enhance navigation and educational engagement within college premises. It provides an augmented reality (AR) experience to guide users through campus locations, a chatbot for real-time assistance, and a learning platform built on Django for educational content access. The system targets students, visitors, and faculty, offering seamless navigation, real-time help, and secure access to learning resources. Key features include AR markers for location guidance, a responsive user interface, and robust data security measures. The platform aims to improve the user experience by integrating AR, chatbot, and learning functionalities into a unified system.

Chapter 1

Introduction

1.1 Purpose

The purpose of the AR Navigation System with Chatbot and Learning Platform is to provide a comprehensive digital solution that enhances the user experience on a college campus. It aims to:

1. **Facilitate Navigation**: Assist students, visitors, and faculty in navigating the campus using AR technology, providing real-time directions and location guidance.
2. **Offer Real-Time Assistance**: Integrate a chatbot to answer user queries related to navigation and learning resources, improving overall accessibility and support.
3. **Enhance Learning Engagement**: Provide a platform for accessing educational content, managing courses, and tracking academic progress, thereby enriching the learning environment.
4. **Improve Campus Accessibility**: Ensure easy access to information and locations within the campus, promoting a user-friendly and efficient experience for all users.
5. **Promote Secure Data Usage**: Protect user data with strong authentication, encryption, and privacy measures, ensuring a safe digital environment.

1.2 Problem Statement

Navigating large and complex college campuses can be challenging for students, visitors, and faculty members, leading to confusion, lost time, and frustration. Existing navigation methods, such as static maps and directional signs, are often insufficient, especially for newcomers. Additionally, accessing learning materials and obtaining real-time assistance on campus is often fragmented across different platforms, making it inconvenient for users to find the information they need promptly.

The current system lacks an integrated solution that combines AR-based navigation, immediate support through chatbots, and a centralized learning management platform. This gap results in inefficient navigation, lack of immediate help, and poor accessibility to educational resources, negatively impacting the overall campus experience.

Therefore, there is a need for a comprehensive digital solution that provides seamless navigation, real-time assistance, and integrated learning resources, tailored specifically for the campus environment. The AR Navigation System with Chatbot and Learning Platform addresses these challenges by offering an innovative, user-friendly approach to enhancing campus navigation and learning experiences.

1.3 Objectives

- To develop an intuitive AR-based navigation system that guides users accurately through indoor and outdoor campus locations using AR markers and GPS integration.
- To implement a chatbot that provides instant responses to user queries related to navigation, campus facilities, and learning resources
- To create a learning platform where students can access courses, study materials, and track their academic progress, managed by a user-friendly interface.
- To establish robust authentication methods to ensure secure access to the system, protecting user data and adhering to privacy standards.

1.4 Scope

- Can be used in the Educational sector: The system includes a learning platform and aims to enhance the educational experience, clearly indicating its applicability in the educational sector.
- Can be useful to Students and Teachers: The system caters to both students (through navigation, chatbot assistance, and the learning platform) and faculty (who can manage courses and content), making it beneficial for both groups.
- Can act as a reliable solution toward the problem of mismatched/misplaced notes: The learning platform allows users to access courses and view content, which can include notes and other study materials. This centralized access can help mitigate issues related to mismatched or misplaced physical notes.
- Can provide real-time assistance: The chatbot functionality offers immediate responses to user queries, enhancing the user experience and providing on-demand support.
- Can enhance the learning experience: The platform hosts learning resources, enabling users to access courses, submit assignments, and track their progress, contributing to a more engaging and effective learning environment.
- Can enable seamless navigation: The AR-based navigation system facilitates easy indoor and outdoor navigation, particularly useful within a large campus or complex building layouts.
- Can prioritize user data security and privacy: The system incorporates robust authentication mechanisms and data encryption to protect user information.
- Can offer accessibility across devices: The system is designed to be mobile-first, ensuring optimal usability on smartphones and tablets while also being accessible on desktops.
- Can facilitate effective search functionality: The APSITwiki within the system includes search options, enabling users to quickly locate relevant information.

Chapter 2

Literature Review

Literature 1

Web AR: A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights

The reference text provides a comprehensive overview of Web AR, a promising technology that leverages the capabilities of the web to deliver augmented reality experiences on mobile devices.

What is Web AR?

Web AR refers to the implementation of augmented reality (AR) experiences directly within web browsers, eliminating the need for dedicated apps. It allows users to access and interact with AR content seamlessly across various devices and platforms.

Key Features and Functions:

- **Cross-Platform Accessibility:** Web AR leverages the inherent cross-platform nature of the web, enabling AR experiences to be accessed on a wide range of devices and operating systems without requiring separate app development.
- **Lightweight Implementation:** Web AR solutions prioritize efficiency and minimize resource consumption, making them suitable for deployment on mobile devices with limited computing power and battery life.
- **Seamless Integration:** Web AR seamlessly blends virtual content with the real world, enhancing user perception and interaction with their surroundings.
- **Diverse Applications:** Web AR has the potential to revolutionize various sectors, including education, entertainment, advertising, navigation, and maintenance, by providing immersive and interactive experiences.

Challenges and Enabling Technologies:

- **Limited Computing Capability:** The computationally intensive nature of AR poses challenges for web browsers, requiring efficient algorithms and techniques to ensure smooth performance on mobile devices.
- **Network Delay:** Real-time AR experiences demand low-latency communication, making network efficiency and optimization crucial for Web AR applications.
- **Energy Consumption:** The continuous use of sensors, processing, and communication in AR applications can strain the battery life of mobile devices, necessitating energy-efficient solutions.
- **Compatibility:** The diversity of web browsers, devices, and 3D model formats can create compatibility challenges for Web AR development and deployment.

- **Privacy and Security:** Web AR applications often involve collecting and processing user data, raising concerns about privacy and security that need to be addressed.

Benefits:

- **Pervasive Promotion:** Web AR's accessibility and ease of use can facilitate the widespread adoption of AR technology, reaching a broader audience.
- **Enhanced User Experience:** Web AR can deliver immersive and interactive experiences that enrich user interactions with the real world.
- **Cost-Effectiveness:** By eliminating the need for dedicated app development and leveraging existing web infrastructure, Web AR can offer cost-effective solutions for AR deployment.

In summary, Web AR represents a promising future for mobile augmented reality, offering a lightweight, cross-platform, and pervasive approach to delivering AR experiences. While challenges remain in terms of computing efficiency, network optimization, energy consumption, and compatibility, ongoing research and development efforts are paving the way for the widespread adoption and transformative impact of Web AR across various domains.

Literature 2

A Systematic Review of Application Development in Augmented Reality Navigation Research

This review explores the landscape of application development within augmented reality navigation research. It specifically investigates trends in hardware, software, and methodologies employed in this field.

Key Findings and Functions:

- **Growth in AR Navigation Research:** The review highlights a significant increase in interest in AR navigation research, particularly since 2010. This surge is attributed to the widespread adoption of smartphones and advancements in AR technology.
- **Inconsistency in Technical Reporting:** A key concern identified is the lack of consistency in how technical details are reported across AR navigation research studies. This inconsistency poses challenges for comparison and synthesis of findings.
- **Focus on Trends:** To address this gap, the review undertakes a comprehensive analysis of trends in hardware, software, and methodologies utilized in AR navigation applications.
- **Implications for Future Research:** The insights gleaned from this review serve as a valuable foundation for future research endeavors in AR navigation. It emphasizes the

need for standardized reporting practices to facilitate meaningful comparisons and knowledge accumulation.

Benefits:

- **Comprehensive Overview:** The review provides a comprehensive overview of the current state of AR navigation research.
- **Identifies Key Trends:** It identifies key trends in hardware, software, and methodologies, offering valuable insights for researchers and developers.
- **Highlights Areas for Improvement:** The review underscores the need for standardized reporting practices, promoting transparency and comparability.

This systematic review plays a crucial role in advancing AR navigation research by providing a structured analysis of trends and highlighting areas for improvement. It serves as a valuable resource for researchers, developers, and practitioners interested in the field.

Literature 3

Indoor Navigation Using Augmented Reality

This article explores the potential of augmented reality (AR) for indoor navigation. It presents an AR-based indoor navigation system that leverages QR codes, 3D building models, and a shortest path algorithm to guide users within indoor environments.

Key Findings and Functionalities:

- **Addressing GPS Limitations:** The article highlights the limitations of GPS technology in indoor environments, where signal reception is often weak or unavailable. AR navigation emerges as a promising solution to overcome these limitations.
- **Leveraging AR Technology:** The proposed system utilizes AR to superimpose virtual directions onto the user's real-world view through their smartphone camera. This provides intuitive and clear guidance for navigation.
- **QR Code Integration:** QR codes strategically placed within the indoor environment serve as reference points for the system to accurately track the user's location.
- **3D Model Utilization:** A 3D model of the building is employed to create a comprehensive map of the indoor space, enabling the system to calculate optimal routes.
- **Shortest Path Algorithm:** The system implements a shortest path algorithm to determine the most efficient path from the user's current location to their desired destination.

Benefits:

- **Enhanced Indoor Navigation:** The AR-based system offers a more accurate, reliable, and user-friendly navigation experience compared to traditional methods in GPS-deprived environments.
- **Improved User Experience:** The visual guidance provided through AR fosters a more intuitive and engaging navigation experience for users.
- **Increased Accessibility:** This technology holds significant potential to benefit individuals with visual impairments or those unfamiliar with the indoor environment.

In summary: This article presents a novel and promising approach to indoor navigation using augmented reality. The system's ability to address GPS limitations, leverage AR technology, and provide efficient navigation makes it a valuable contribution to the field. Future research could explore further integration with other technologies and investigate its broader applicability in various indoor settings.

Chapter 3

Proposed System

3.1 Features and Functionality

1. AR Navigation:

This innovative AR navigation system centralizes all location information into a streamlined, coordinate-based approach. As you move throughout the campus, the system dynamically guides you to your chosen destination using real-time updates and AR overlays, ensuring you're always on the right track. Whether navigating indoors or outdoors, this system seamlessly provides intuitive directions without relying on physical markers or maps. It's like having a personal navigation assistant right on your smartphone, making even the most complex campuses easy to explore.

2. Chatbot:

Streamlines access to campus knowledge, directly from your dashboard. Real-Time Assistance serves as a central hub for all your campus inquiries, offering immediate responses to questions about navigation, campus information, and learning resources – all conveniently located on your dashboard. It functions like a website's homepage, providing a clear starting point for accessing the information you need. Frequently asked questions are addressed efficiently through predefined answers, similar to a quick links section. While the chatbot is readily available on your dashboard, it's important to note that it's not directly accessible within the navigation or learning platforms themselves. This design ensures a dedicated space for seeking assistance, fostering a streamlined and focused approach to accessing campus knowledge.

3. Learning Platform:

APSITwiki is a specific wiki dedicated to A. P. Shah Institute of Technology (APSIT). Its purpose is to serve as a centralized academic repository, containing study materials, lecture notes, and textbooks. Additionally, APSITwiki encourages collaboration among students, professors, and alumni. It goes beyond traditional classroom learning by offering project resources, career insights, and practical applications. In summary, APSITwiki empowers learners by providing easy access to essential information and promoting knowledge sharing

4. Search and Filtering:

Search enables advanced options to quickly locate courses, notes, and other essential learning materials. It’s like having a site-wide search function, helping you pinpoint the exact information you need. On the other hand, Location Search acts as a directory, allowing you to pinpoint specific locations like classrooms or faculty offices within the campus – just as you'd use a search bar to find specific pages on a website. These search functionalities provide a clear path for students to navigate the vast landscape of academic resources and physical spaces, ensuring they can find what they need when they need it.

Chapter 4

Technical Specification

1.Python Language:

Python is a high-level, interpreted programming language known for its easy-to-read syntax and dynamic semantics. It's widely used for Rapid Application Development and as a scripting language to connect existing components. Here are some key points:

- Interpreted Language: Python code is executed line by line, making debugging easier and eliminating the need for a separate compilation step.
- Dynamic Typing: Variables in Python can change type dynamically, which adds flexibility to the language.
- Built-in Data Structures: Python comes with ready-to-use data structures, which simplifies code and enhances productivity.
- Extensive Libraries: A vast standard library and modules support various programming tasks, from web development to data analysis.

2.Django Framework to develop GUI:

Django is our framework of choice to build the GUI with since it offers a diverse range of built-in methods and parameters which expertly advance our user interface in usability.Django is a high-level Python web framework that empowers developers to build robust and maintainable web applications efficiently. Here are the key points about Django:

- Philosophy and Purpose:
  - Django simplifies web development by handling repetitive tasks, allowing developers to focus on writing their applications.
  - It adheres to the DRY (Don't Repeat Yourself) principle, emphasizing automation and code reusability.

- Features and Benefits:
  - Rapid Development: Django accelerates project implementation, reducing the time needed to build web applications.
  - Secure: Built by experienced developers, Django incorporates security best practices.
  - Scalable: It handles scalability challenges, making it suitable for both small projects and large-scale applications.
  - Fully Loaded: Django includes built-in features like authentication, admin interface, and database management.

    - Versatile: Developers can create various types of applications, from content management systems to social networks.
    - Open Source: Django is free and open source, with an active community contributing to its growth.
- MVT Design Pattern:
    - Django follows the MVT (Model-View-Template) design pattern:
        - Model: Represents the data structure (database schema).
        - View: Handles data processing and interacts with templates.
        - Template: Defines how data is presented in HTML.
- History and Popularity:
    - Django was publicly released in 2005 and has since gained widespread adoption.
    - Notable sites like Instagram, Mozilla, and Pinterest use Django for their web applications.

In summary, Django streamlines web development, provides security, and encourages clean, pragmatic design. Developers can build powerful applications while relying on Django's robust features

3. Visual Studio Code(VCS) as Integrated Development Environment (IDE):

Visual Studio Code (VCS) is a very specific pick for our project as it provides a very speculative look into our code along with quick and responsive info which assisted us in launching forward with a headstart on the frontend creation. Visual Studio is a powerful Integrated Development Environment (IDE) developed by Microsoft. It's designed for building a wide range of applications, including desktop, web, mobile, and cloud services. Here's a brief overview:

- **Multi-Language Support**: Visual Studio supports multiple programming languages like C#, C++, VB, Python, and JavaScript.
- **Development Platforms**: It uses Microsoft software development platforms like Windows API, Windows Forms, and Windows Presentation Foundation (WPF).
- **Editions**: Available in Community, Professional, and Enterprise editions, with the Community version being free for individual developers and open-source projects.
- **Features**: Includes a code editor with IntelliSense, code refactoring, debugging, and many other tools for software development.

4. HTML:

HTML, or HyperText Markup Language, is the standard markup language for creating web pages. It structures web content and is often assisted by CSS (Cascading Style Sheets) and JavaScript. Here's a brief overview:

- Markup Language: HTML uses tags to define the structure and formatting of web content.
- Elements and Tags: Elements are the building blocks of HTML pages, enclosed by tags like <html>, <head>, and <body>.
- Web Browsers: Browsers interpret HTML documents to render multimedia web pages.
- HTML5: The latest version, HTML5, supports multimedia elements like audio and video.

5. CSS:

CSS, or Cascading Style Sheets, is a style sheet language used to describe the presentation of a document written in HTML or XML. It controls how web pages are displayed in browsers, allowing you to style elements using selectors and declarations. Here's a brief overview:

- Presentation Control: CSS manages layout, colors, fonts, and overall look of web content.
- Separation of Content: It separates content from presentation, improving accessibility and maintainability.
- Flexibility: CSS enables different styles for various devices and screen sizes.
- Versions: The latest version, CSS3, introduced features like animations, transitions, and grid layouts.

6. Markdown:

Markdown is a lightweight markup language created by John Gruber in 2004. It's designed to be easy to write and read, with a simple syntax that can be converted to HTML. Here's a brief overview:

- Plain Text Formatting: Markdown allows you to format text using simple characters like asterisks for bold or underscores for italics.
- Extensibility: It supports additional features like tables, code blocks, and lists, making it versatile for various documentation needs.
- Widely Used: Popular on platforms like GitHub, Stack Overflow, and many forums for its readability and ease of use.
- File Extension: Markdown files typically have the extension .md or .markdown.

7. A-Frame.js:

A-Frame.js is a powerful and intuitive web framework designed to simplify the creation of immersive 3D and virtual reality (VR) experiences directly in your web browser. Built on top of the popular Three.js 3D library, A-Frame provides a declarative, HTML-like syntax that makes it remarkably easy for developers to build interactive 3D scenes without the need for extensive JavaScript knowledge. By utilizing familiar HTML tags and attributes, A-Frame democratizes the creation of VR and AR content, enabling both seasoned developers and newcomers to craft captivating experiences that run seamlessly across various platforms and devices.

- **Declarative Syntax:** A-Frame leverages an HTML-like structure, employing custom elements and attributes to define 3D objects, lighting, materials, animations, and interactions within a scene. This declarative approach promotes readability and maintainability, making it easier to understand and modify the code.
- **Entity-Component System (ECS):** A-Frame adopts an ECS architecture, where entities represent objects in the scene, and components define their properties and behaviors. This modular structure enhances flexibility and reusability, allowing developers to easily add, remove, or modify components to customize the functionality of their 3D objects.
- **Cross-Platform Compatibility:** A-Frame experiences are designed to run natively in web browsers, eliminating the need for additional plugins or installations. This ensures broad accessibility across various devices, including desktops, laptops, smartphones, and VR headsets.
- **VR and AR Support:** A-Frame seamlessly integrates with WebXR, the open standard for web-based VR and AR experiences. This allows developers to create immersive content that users can interact with using VR headsets or AR-enabled devices.
- **Community and Ecosystem:** A-Frame boasts a vibrant and supportive community of developers, along with a rich ecosystem of pre-built components and plugins. This wealth of resources facilitates rapid development and encourages collaboration among creators.
- **Ease of Use:** A-Frame's beginner-friendly approach makes it an excellent choice for those new to 3D development or VR/AR. Its intuitive syntax and extensive documentation empower developers to quickly get started and build impressive experiences without a steep learning curve.
- **Powerful Features:** Despite its simplicity, A-Frame offers a wide range of advanced features, including physics simulations, animation tools, and event handling, enabling developers to create sophisticated and interactive 3D scenes.

A-Frame.js is a versatile and accessible web framework that empowers developers to create immersive 3D and VR/AR experiences directly within their web browsers. Its declarative syntax, ECS architecture, and cross-platform compatibility make it an excellent choice for both beginners and experienced developers looking to harness the potential of web-based 3D content. With its thriving community and extensive ecosystem, A-Frame offers a wealth of resources and support to help creators bring their 3D visions to life.

8. Three.js:

Three.js is a powerful and widely-used JavaScript 3D library that simplifies the creation of stunning 3D graphics and animations directly within the web browser. It provides a comprehensive set of tools and functionalities for rendering 3D scenes, handling user interactions, and incorporating various visual effects. With its flexible and extensible architecture, Three.js empowers developers to build a wide range of 3D applications, from interactive visualizations and games to virtual reality (VR) and augmented reality (AR) experiences.

- **Scene Graph:** Three.js employs a scene graph structure to represent the 3D world, organizing objects, lights, cameras, and other elements in a hierarchical manner. This structure facilitates efficient rendering and enables complex scene management.
- **WebGL Renderer:** Three.js leverages WebGL, a web standard for rendering 3D graphics, to tap into the hardware acceleration capabilities of modern GPUs. This delivers impressive performance and visual quality, even for complex scenes.
- **Object Representation:** Three.js provides a variety of classes for representing different types of 3D objects, including meshes, geometries, materials, and textures. This allows developers to construct a wide range of 3D models and apply realistic visual effects.
- **Lighting and Shadows:** Three.js supports various lighting models, including ambient, directional, point, and spot lights, to illuminate scenes and create realistic shadows. This contributes to the visual depth and realism of 3D environments.
- **Animation and Interaction:** Three.js offers built-in animation capabilities, allowing developers to animate object properties, cameras, and other elements over time. Additionally, it provides tools for handling user interactions, such as mouse clicks and keyboard input, enabling interactive 3D experiences.
- **Camera Control:** Three.js includes camera controls for navigating and exploring 3D scenes. This enables users to pan, zoom, and rotate the camera to view objects from different perspectives.
- **Extensibility:** Three.js boasts a modular and extensible architecture, allowing developers to create custom geometries, materials, shaders, and other components. This provides flexibility and adaptability for specific project requirements.
- **Community and Ecosystem:** Three.js has a large and active community of developers, along with a wealth of tutorials, examples, and plugins. This supportive ecosystem fosters collaboration and knowledge-sharing among creators.

Three.js is a versatile and powerful 3D library that enables developers to build stunning and interactive 3D graphics directly in the web browser. Its extensive feature set, performance capabilities, and active community make it a popular choice for creating a wide range of 3D applications, from games and visualizations to VR and AR experiences. Whether you're a seasoned 3D developer or just starting your journey, Three.js provides the tools and resources you need to bring your creative visions to life in the 3D world**.**

Chapter 5

Project Design

5.1 Use Case diagram

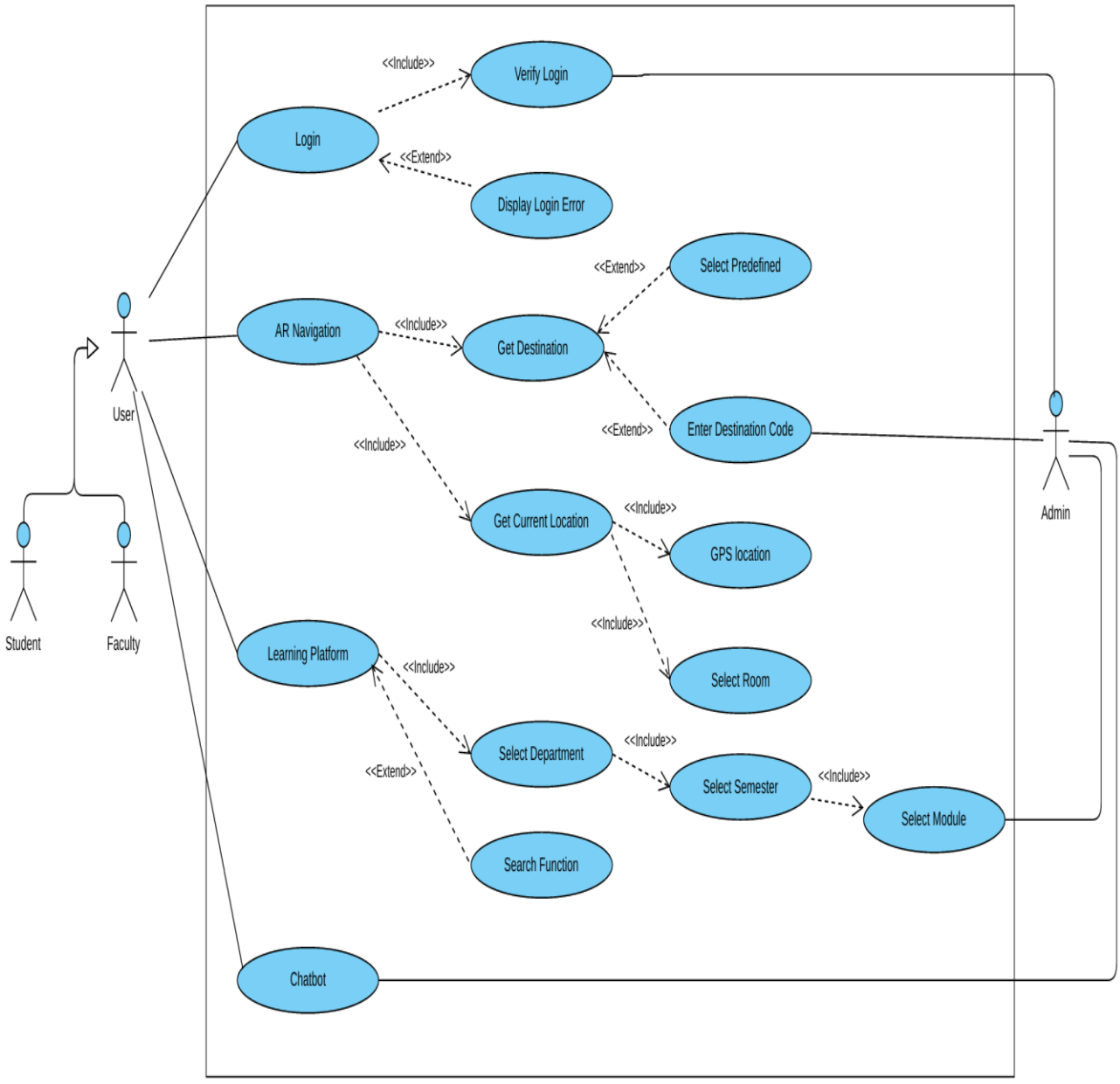


Fig 5.1 : Use Case Diagram

This use case diagram outlines the interactions between various users (Student, Faculty, Admin) and a system that likely represents an educational institution's platform with integrated AR navigation and learning capabilities.

Key Components:

- **Actors:**
    - **User:** A general user who can log in, potentially accessing AR navigation and other features.
    - **Student:** Interacts with the Learning Platform for academic purposes.
    - **Faculty:** Likely manages courses, content, and assessments within the Learning Platform.
    - **Admin:** Manages the system, potentially handling user accounts, content, and settings.
- **Use Cases:**
    - **Login:** Users authenticate themselves to access the system.
        - **Verify Login:** Checks credentials and grants access if valid.
        - **Display Login Error:** Indicates invalid credentials.
    - **AR Navigation:**
        - **Get Destination:** The system obtains the user's desired destination.
        - **Enter Destination Code:** Allows manual input of a destination code.
        - **Get Current Location:** The system determines the user's current location, potentially using GPS.
    - **Learning Platform:**
        - **Select Predefined:** Users choose from pre-set options, possibly related to courses or resources.
        - **Search Function:** Enables searching for specific content or information within the platform.
    - **Admin:**
        - **GPS location:** Admin may have access to track or manage GPS locations (potentially for AR navigation or other purposes).
        - **Select Room:** Admin might select specific rooms for management or configuration.
        - **Select Department:** Admin might manage or view information related to departments.

        - **Select Semester:** Admin might manage or view information related to academic semesters.
        - **Select Module:** Admin might manage or view information related to course modules.

- **Relationships:**
    - **Includes:** Indicates that one use case is a necessary part of another. For instance, "Get Current Location" is included in "AR Navigation."
    - **Extends:** Suggests an optional or conditional extension of a use case.

Overall Functionality:

- **User Authentication:** Users must log in to access the system.
- **AR Navigation:** Users can find their way around using AR technology, either by selecting a predefined destination or entering a code. The system uses their current location (likely via GPS) to guide them.
- **Learning Platform:** Students and Faculty interact with the Learning Platform, accessing course materials and resources.
- **Admin Functions:** The Admin has additional capabilities for managing the system, including potential oversight of GPS locations, rooms, departments, semesters, and modules.
- **Chatbot:** Available for general assistance or inquiries throughout the platform.

5.2 Flow Diagram

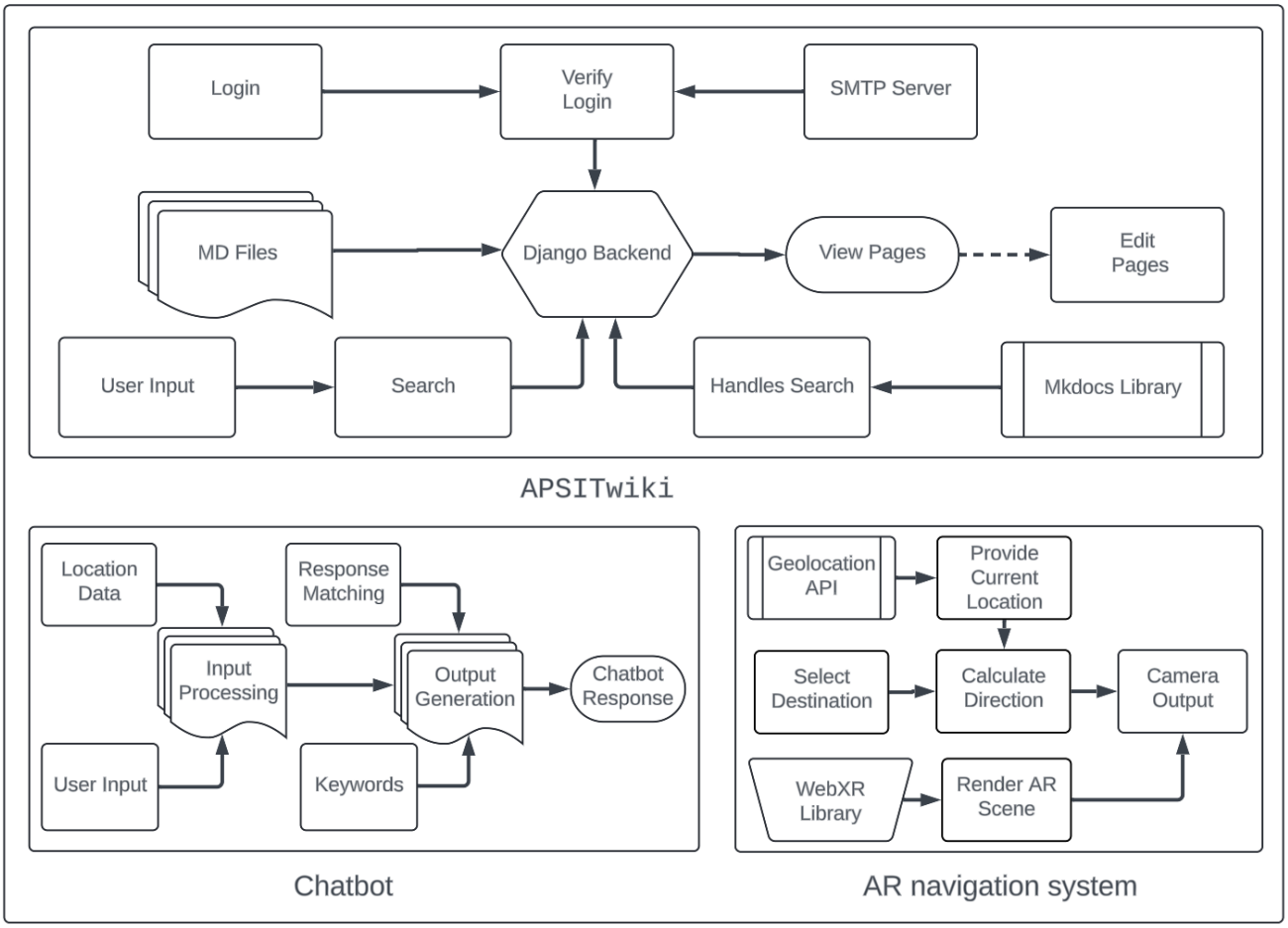


Fig 5.2: Data Flow Diagram

Components:

- **Login:** This is the entry point for users to access the system.
- **Verify Login:** This process checks the user's credentials against a database or authentication system.
- **SMTP Server:** This server is responsible for sending email notifications, likely related to user registration, password resets, or other system-generated messages.
- **MD Files:** These are Markdown files, a lightweight markup language often used for writing documentation and web content.
- **Django Backend:** This is the core of the system, built using the Django web framework. It handles data storage, retrieval, and processing.
- **View Pages:** Users can view the rendered content (web pages) generated from the Markdown files.

- **Edit Pages:** Authorized users can edit the Markdown files to update or create new content.
- **User Input:** This represents any input provided by the user, such as search queries.
- **Search:** This component processes user search queries.
- **Handles Search:** The Django backend likely handles the search functionality, retrieving relevant content from the MD files or database.
- **Mkdocs Library:** This is a static site generator that converts Markdown files into a structured website.
- **User Input**: The process starts with the user providing input, likely in the form of text or voice commands.
- **Input Processing**: This component takes the user's input and analyzes it. This could involve natural language understanding (NLU) tasks like tokenization, part-of-speech tagging, and entity recognition to extract meaning from the input.
- **Keywords**: Keywords or key phrases relevant to the user's query are identified during input processing.
- **Location Data:** The system also receives location data, possibly from the user's device's GPS or other location services.
- **Response Matching**: This stage involves matching the extracted keywords and location data with relevant information or responses stored in the system's knowledge base or database.
- **Output Generation:** Once a match is found, this component generates a suitable response. This might involve natural language generation (NLG) techniques to create a coherent and contextually relevant answer.
- **Chatbot Response**: The final generated response is delivered to the user through the chatbot interface.
- **Geolocation API:** This component likely interacts with the device's GPS or other location services to obtain the user's current location.
- **Provide Current Location:** This process utilizes the Geolocation API to determine the user's real-time location.
- **Select Destination:** The user inputs or selects their desired destination within the system.
- **Calculate Direction:** Based on the user's current location and the selected destination, this component calculates the route or directions to reach the destination.
- **WebXR Library:** This library is responsible for creating and managing web-based Augmented Reality (AR) experiences.
- **Render AR Scene:** This component utilizes the WebXR library to generate and display AR elements (e.g., navigation arrows, directions, points of interest) overlaid onto the real-world view captured by the camera.
- **Camera Output:** This represents the live feed from the device's camera, which serves as the background for the AR scene.

Data Flow:

1. **User Authentication:**
    - The user provides login credentials.
    - The system verifies the login, possibly interacting with an external authentication service or database.
    - If successful, the user is granted access to view and potentially edit content.
    - The SMTP server may be used to send confirmation or notification emails.
2. **Content Management:**
    - MD Files store the raw content in Markdown format.
    - The Django Backend interacts with the MD files, storing and retrieving data as needed.
    - Mkdocs Library converts Markdown files into HTML web pages.
    - Users can view the rendered pages.
    - Authorized users can edit the MD files, which triggers an update to the rendered pages.
3. **Search Functionality:**
    - The user provides input (search query).
    - The Search component processes the query.
    - The Django Backend handles the search, retrieving matching content from the MD files or database.
    - The search results are presented to the user.
4. **AR Navigation:**
    - The system uses the Geolocation API to get the user's current location.
    - The user specifies their intended destination.
    - Using the current location and destination, the system calculates the optimal route or directions.
    - The WebXR library creates an AR scene, incorporating the calculated directions and other relevant information.
    - The AR scene is superimposed onto the live camera feed, providing the user with real-time AR navigation guidance.

5.3 System Architecture

Fig 5.3: System Architecture

The system architecture can be explained as follows:

1. Client Side:

- **HTML/CSS/JavaScript:** This represents the frontend of the application, responsible for the user interface and interactions.
- **Chatbot UI:** This is the visual interface through which users interact with the chatbot.
- **AR Navigation:** This component handles the augmented reality navigation experience, likely leveraging device sensors and camera input.

2. Backend Server:

- **Django Framework:** This Python-based web framework powers the backend, handling server-side logic, data processing, and communication with the client-side and external services.
- **Chatbot Logic (Pre-Defined):** This suggests that the chatbot's responses and behavior are primarily rule-based or scripted, rather than relying on complex AI or machine learning.

3. Authentication System:

- **FireBaseAuthentication System:** FireBase is a third-party authentication service that likely handles user registration, login, and session management for the application.

4. Deployment and External Integration:

- **GitHub Pages (for AR):** The AR component of the application is likely hosted and deployed using GitHub Pages, a simple static website hosting service.
- **Third-Party APIs:**
    - **GPS:** The application integrates with a GPS API to access location data for the AR navigation feature.
    - **OAuth :** This indicates the use of OAuth (Open Authorization) protocol for secure authentication and authorization with FireBase service.

5.4 Implementation

APSIT AR Navigation Map is an integrated digital tool designed to enhance the campus experience at A. P. Shah Institute of Technology (APSIT). It combines augmented reality (AR) navigation to help users easily find their way around the campus, a chatbot for real-time assistance, and a learning platform that hosts courses and educational resources. The system also features a searchable wiki, offering quick access to important academic information. In essence, APSIT AR Navigation and Learning Platform simplifies campus navigation, supports learning, and fosters a connected educational environment.

AR Navigation:

The AR Navigation System uses an augmented reality interface to help users navigate campus environments.

To access the navigation feature, users can select a room or location from a dropdown menu. The system then uses GPS coordinates to determine the user's current location and calculate the direction to the selected destination.

Ensure that the location services on your device are enabled and that you have granted the necessary camera permissions. The AR system uses a 3D arrow to guide users by updating its direction in real-time based on the user's movement and orientation.

The system uses a straightforward mechanism where the user's position and destination are mapped using basic vector calculations. The arrow's direction is adjusted accordingly to help the user reach their destination efficiently.

```
let destination = new THREE.Vector3();

function updateDestination(lat, lng) {
    navigator.geolocation.getCurrentPosition(position => {
        const { latitude, longitude } = position.coords;
        destination = latLngToVector3(lat, lng, latitude, longitude);
    }, showError, {
        enableHighAccuracy: true,
        timeout: 5000,
        maximumAge: 0
    });
}

function showError(error) {
    console.error('Geolocation error:', error);
    alert('Error getting your location. Please ensure location services are enabled and try again.');
}
```

Fig 5.4.1: Geolocation API snippet

```
function latLngToVector3(latDest, lngDest, latOrigin, lngOrigin) {
    const earthRadius = 6371; // Earth's radius in kilometers

    const latDiff = THREE.MathUtils.degToRad(latDest - latOrigin);
    const lngDiff = THREE.MathUtils.degToRad(lngDest - lngOrigin);

    const a = Math.sin(latDiff / 2) * Math.sin(latDiff / 2) +
              Math.cos(THREE.MathUtils.degToRad(latOrigin)) * Math.cos(THREE.MathUtils.degToRad(latDest)) *
              Math.sin(lngDiff / 2) * Math.sin(lngDiff / 2);

    const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

    const distance = earthRadius * c;

    const x = distance * Math.cos(lngDiff);
    const y = 0;
    const z = distance * Math.sin(lngDiff);

    return new THREE.Vector3(x, y, z);
}
```

Fig 5.4.2: Calculating distance using Three.js snippet

```
    AFRAME.registerComponent('navigation-helper', {
        tick: function () {
            const camera = document.querySelector('[camera]');
            const arrow = document.querySelector('#arrow');

            if (camera && arrow && destination) {
                const cameraRotation = new THREE.Euler();
                camera.object3D.getWorldRotation(cameraRotation);

                const direction = destination.clone();
                direction.applyEuler(cameraRotation);

                let angle = Math.atan2(direction.x, direction.z);
                angle = THREE.MathUtils.radToDeg(angle);

                arrow.setAttribute('rotation', `-90 ${angle} 0`);
            }
        }
    });
```

Fig 5.4.3: Camera input calculation using Three.js

The camera code in this AR navigation system dynamically adjusts the navigation arrow's direction based on the user's orientation and position. By tracking the camera's real-time rotation and applying it to the arrow's rotation, the system ensures that the virtual guide remains aligned with the user's perspective, providing accurate directional assistance within the augmented reality environment.

Chatbot:

The code implements a simple chatbot that responds to predefined user queries using a dictionary of responses. When a user inputs a message, the *get_response* function converts the input to lowercase and searches for a match in the predefined responses using regular expressions. If a match is found, the corresponding response is returned; otherwise, a default message is provided. The chatbot runs in a loop, continuously interacting with the user until the user types "bye" to end the conversation. This chatbot is designed to assist with basic inquiries related to navigation and location-specific requests in an AR environment.

```
responses = {
    "hi": "Hello! How can I help you today?",
    "how are you": "I'm just a bot, but I'm doing well. How about you?",
    "what is your name": "I'm a chatbot. I don't have a name, but you can call me Chatbot
    "start navigation": "To start navigation, please enter your current location and your
    "nearest restrooms": "Sure! Please allow me access to your current location, and I'll
    "find the main entrance": "The main entrance is marked with a large blue arrow in the
    "landmarks near me": "I can help with that! Let me access your location to show nearb
    "time to reach my destination": "Based on your current location and the destination,
    "shortcut to the library": "Yes, there is a shortcut. Follow the highlighted path in
    "way to the cafeteria": "Sure! Follow the green path in your AR view to reach the caf
    "help finding the conference room": "No problem! Just enter the name or number of the
    "find a charging station": "There are charging stations located throughout the buildi
    "is there an elevator nearby": "Yes, there's an elevator close to you. Follow the yel
    "bye": "Goodbye! Have a great day!"
}
```

Fig 5.4.4: Predefined Chatbot Responses

```
def get_response(user_input):
    # Convert user input to lowercase
    user_input = user_input.lower()

    # Check if the input matches any of the defined responses
    for key in responses:
        if re.search(r'\b' + re.escape(key) + r'\b', user_input):
            return responses[key]

    # Default response if no match is found
    return "Sorry, I don't understand that. Can you please rephrase?"
```

Fig 5.4.5: Keyword Matching

This Python code implements a simple chatbot that responds to predefined user queries. It uses regular expressions to match the user's input with a dictionary of responses, providing information related to navigation, campus amenities, and general greetings. If the input doesn't match any predefined keys, the chatbot offers a default response indicating it doesn't understand.

Dashboard :

This layout is used as a base for dashboard. This gives access to the AR Navigation System, Responsive Chatbot and Learning Platform APSITwiki.

```
<!DOCTYPE html>
<html>
  <head>
    <title>AR-APSIT.ai</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <header>
      <div class="logo">
        <img src="download.png" alt="logo" />
        <h3>AR-APSIT.ai</h3>
      </div>
      <nav>
        <ul class="nav">
          <li>
            <a href="#">
              Solutions
              <svg width="10" height="10" viewBox="0 0 24 24" fill="currentColor">
                <path d="M12 15l-8-8h16z" />
              </svg>
            </a>
            <ul>
              <li><a href="#">Overview</a></li>
              <li><a href="#">IndoorMaps</a></li>
            </ul>
          </li>
          <li>
            <a href="#">
              Resources
              <svg width="10" height="10" viewBox="0 0 24 24" fill="currentColor">
                <path d="M12 15l-8-8h16z" />
              </svg>
            </a>
```

Fig 5.4.6: Dashboard layout snippet-1

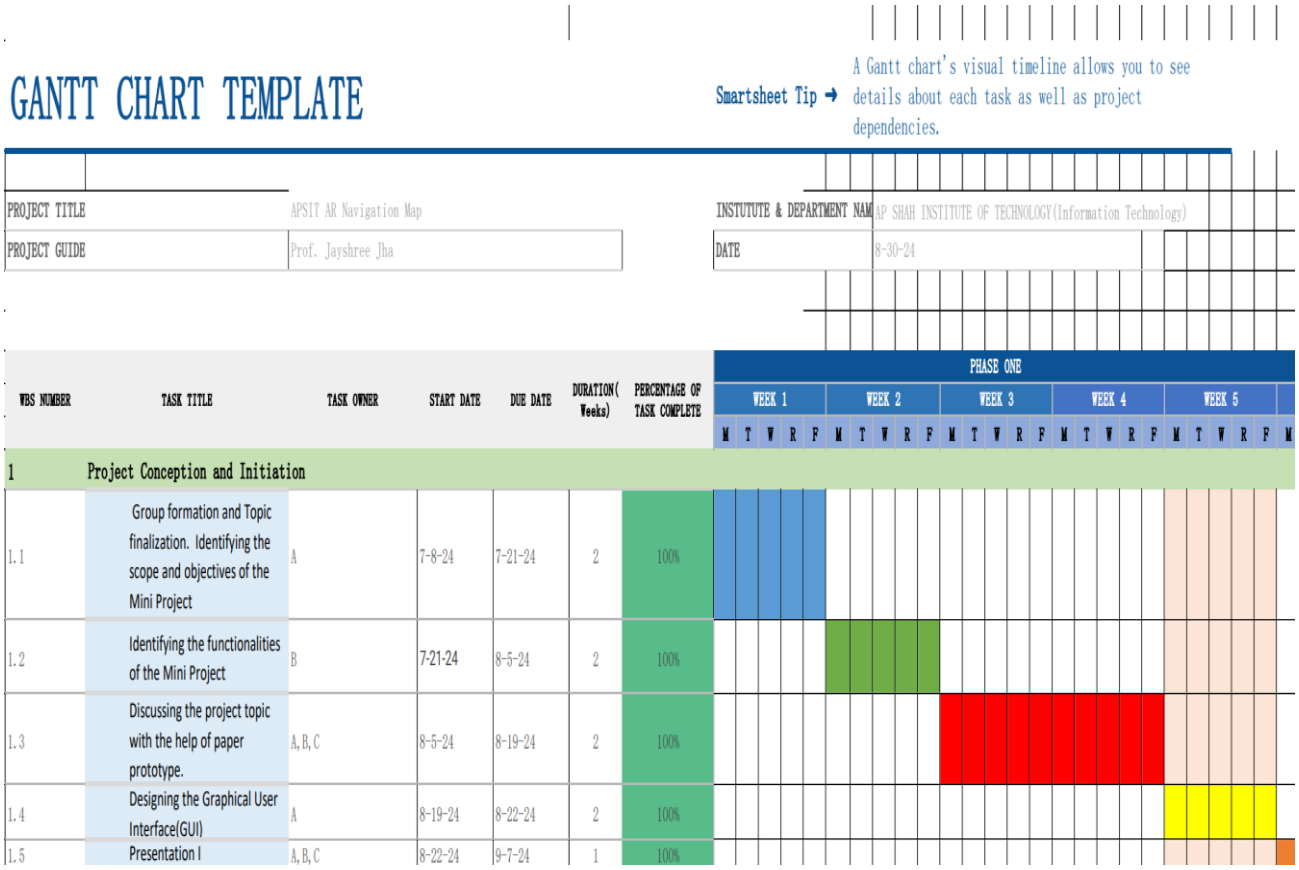
```
<section class="chatbot" onclick="toggleChatbot()">
  <img src="chatbot.png" alt="chatbot" />
</section>

<div class="chatbox" id="chatbox">
  <div class="chatbox-header">Chat with us!</div>
  <div class="chatbox-messages" id="chatbox-messages">
    <div class="question" onclick="sendMessage('hi')">Hi</div><hr>
    <div class="question" onclick="sendMessage('how are you')">How are you?</div><hr>
    <div class="question" onclick="sendMessage('what is your name')">What is your name?</div><hr>
    <div class="question" onclick="sendMessage('start navigation')">Start Navigation</div><hr>
    <div class="question" onclick="sendMessage('nearest restrooms')">Nearest Restrooms</div><hr>
    <div class="question" onclick="sendMessage('find the main entrance')">Find the Main Entrance</div><hr>
    <div class="question" onclick="sendMessage('landmarks near me')">Landmarks Near Me</div><hr>
    <div class="question" onclick="sendMessage('time to reach my destination')">Time to Reach My Destination</div><hr>
    <div class="question" onclick="sendMessage('shortcut to the library')">Shortcut to the Library</div><hr>
    <div class="question" onclick="sendMessage('way to the cafeteria')">Way to the Cafeteria</div><hr>
    <div class="question" onclick="sendMessage('help finding the conference room')">Help Finding the Conference Room</div><hr>
    <div class="question" onclick="sendMessage('find a charging station')">Find a Charging Station</div><hr>
    <div class="question" onclick="sendMessage('is there an elevator nearby')">Is There an Elevator Nearby</div><hr>
    <div class="question" onclick="sendMessage('bye')">Bye</div><hr>
  </div>
</div>
```

Fig5.4.7: Dashboard layout snippet-2

The HTML code represents a webpage for "AR-APSIT.ai," showcasing an AR navigation system. The page features a header with navigation links, a hero section highlighting the system's capabilities, a section showcasing its features, a chatbot for user interaction, and a demo section. Overall, the page aims to introduce users to AR-APSIT.ai and its potential to enhance indoor navigation through augmented reality technology.

Chapter 6
Project Scheduling



GANTT CHART TEMPLATE

Smartsheet Tip → A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

PROJECT TITLE	APSIT AR Navigation Map		INSTUTUTE & DEPARTMENT NAM	AP SHAH INSTITUTE OF TECHNOLOGY(Information Technology)	
PROJECT GUIDE	Prof. Jayshree Jha		DATE	8-30-24	

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION(Weeks)	PERCENTAGE OF TASK COMPLETE
						PHASE ONE
						WEEK 1 WEEK 2 WEEK 3 WEEK 4 WEEK 5
						M T W R F M T W R F M T W R F M T W R F M T W R F M
1	Project Conception and Initiation					
1.1	Group formation and Topic finalization. Identifying the scope and objectives of the Mini Project	A	7-8-24	7-21-24	2	100%
1.2	Identifying the functionalities of the Mini Project	B	7-21-24	8-5-24	2	100%
1.3	Discussing the project topic with the help of paper prototype.	A, B, C	8-5-24	8-19-24	2	100%
1.4	Designing the Graphical User Interface(GUI)	A	8-19-24	8-22-24	2	100%
1.5	Presentation I	A, B, C	8-22-24	9-7-24	1	100%

Fig 6.1 Project Scheduling on weekly basis

Chapter 7

Results

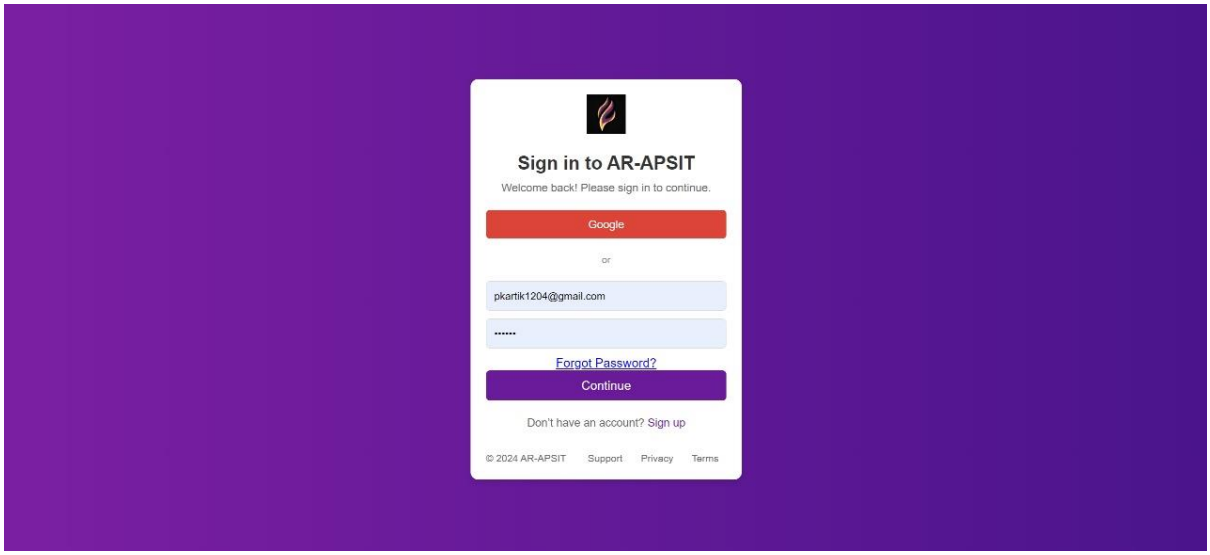


Fig 7.1: SIGN IN Page

- This is the First Page the users sees when they click login on Dashboard page.
- Here the user can enter the login credentials and can access the site.
- The Login Page ensures that only the correct users can access their respective profiles.

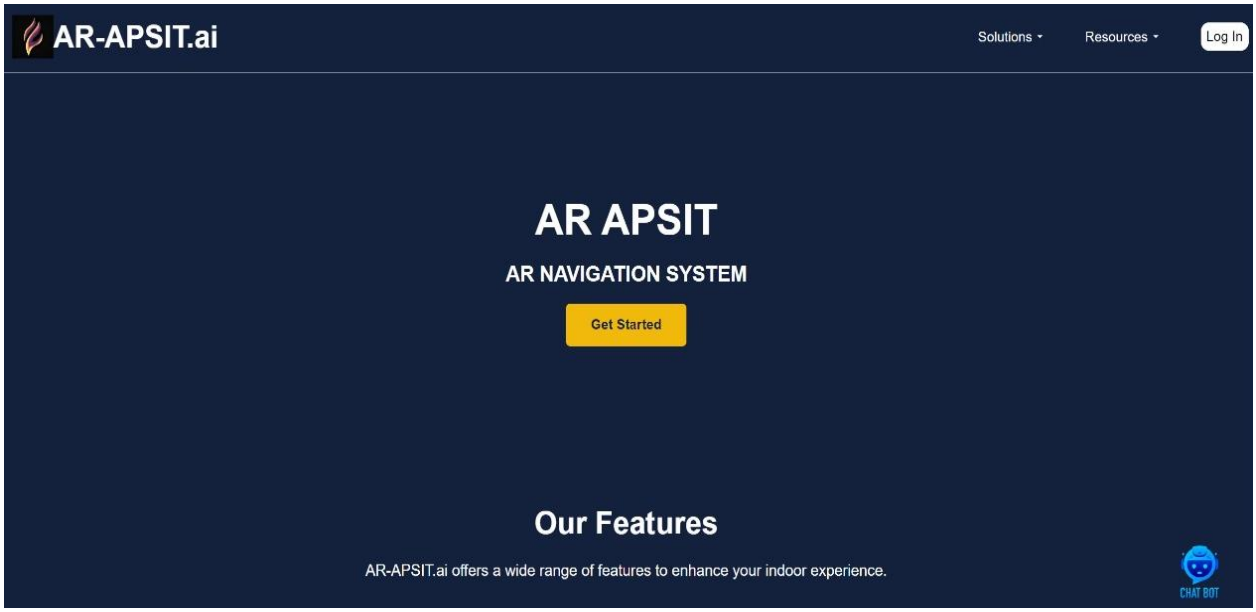


Fig 7.2: DASHBOARD PAGE

- This is the Home Page where we can see a Get Started button which will take us to the options which are AR Navigation and APSITwiki Learning Platform.
- First option is AR Navigation.
- Second option is to APSITwiki Learning Platform.

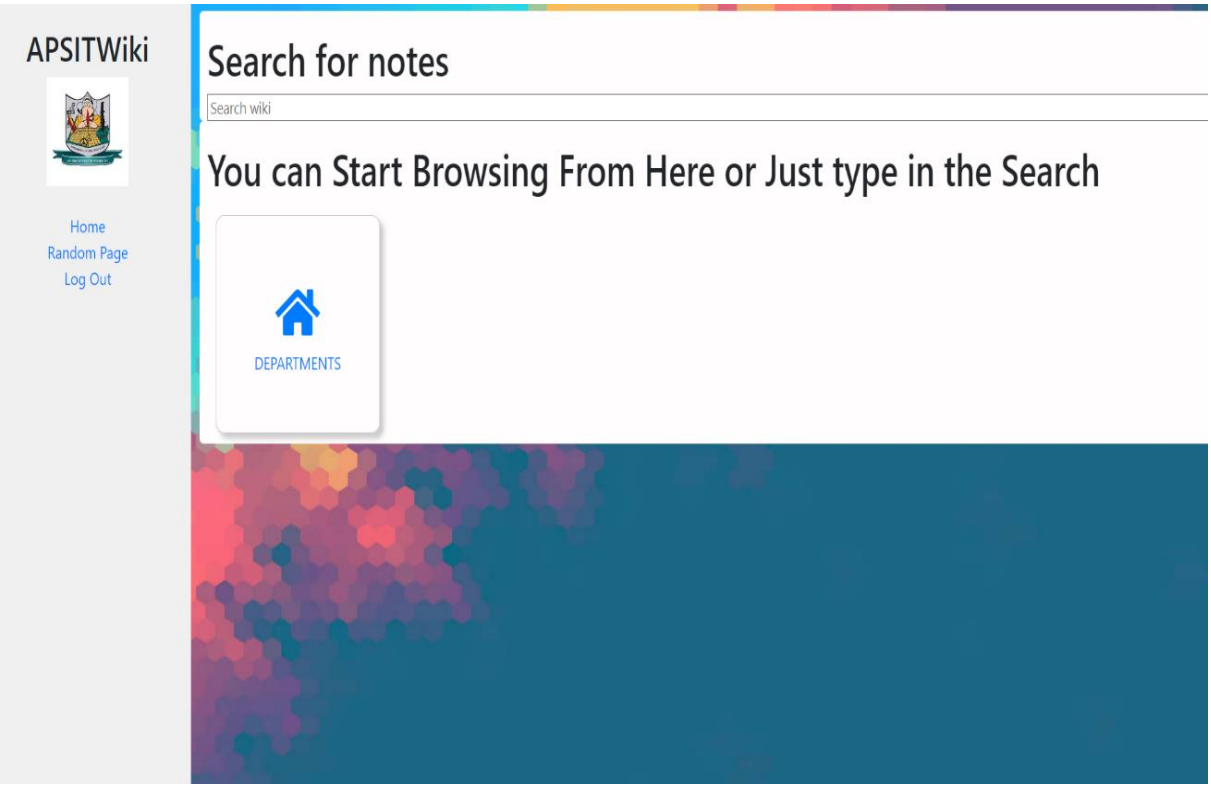


Fig 7.3: APSITwiki HOME PAGE

- Here we can see two options for accessing the notes.
- First option is to browse through the departments
- Second option is to search directly for notes

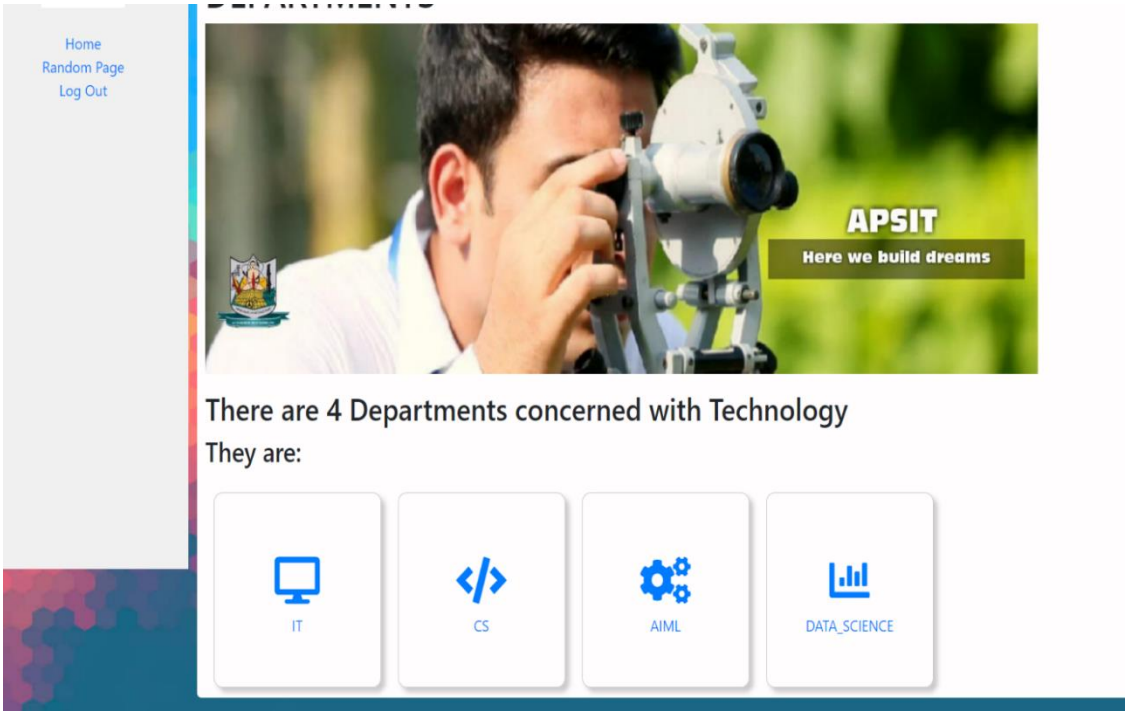


Fig 7.4: NAVIGATING THROUGH DEPARTMENT PAGE

- This page shows different departments so that users can go to their respective departments and access further their notes according to the subjects and modules.
- User can select from various departments.
- User can select one of their respoective departments and access notes from that deaprtment.
- User can then further access the subjects and furthermore user can access their notes.

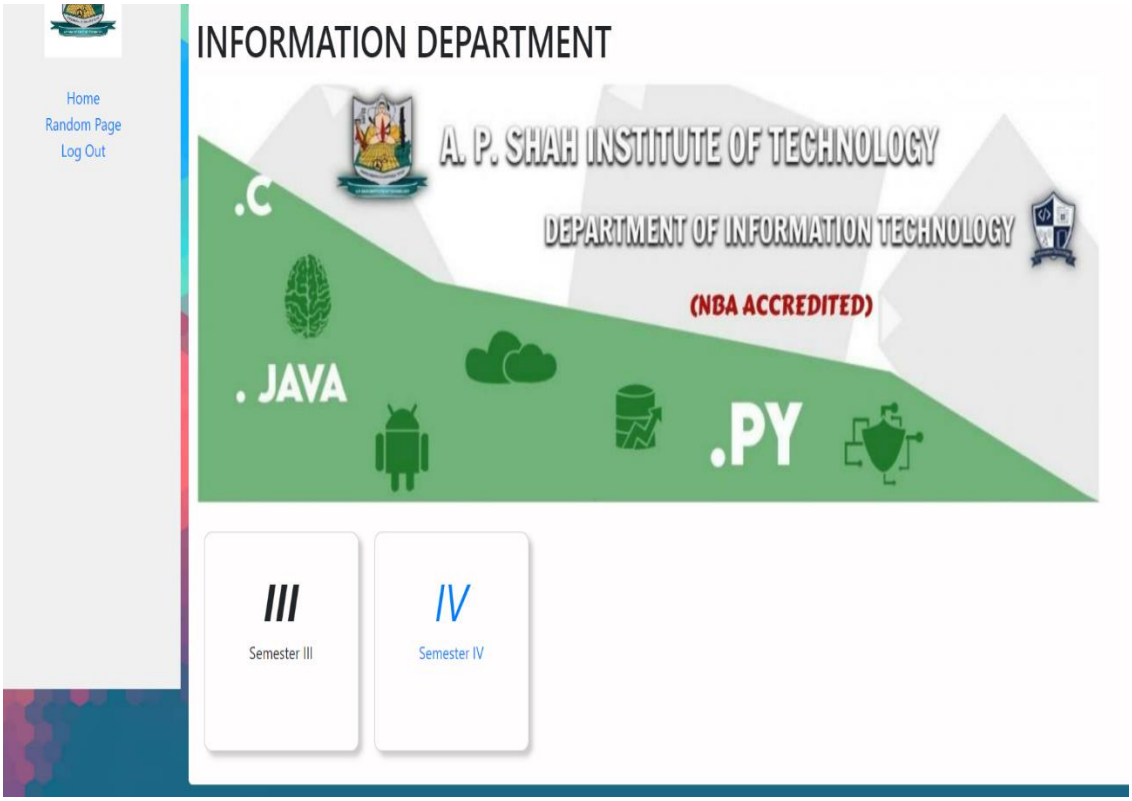


Fig 7.5: INFORMATION TECHNOLOGY PAGE

- This is the Departments page.
- Here user will see the Department Banner
- User can select the semester they want to access
- Furthermore the User can access notes of the semester.

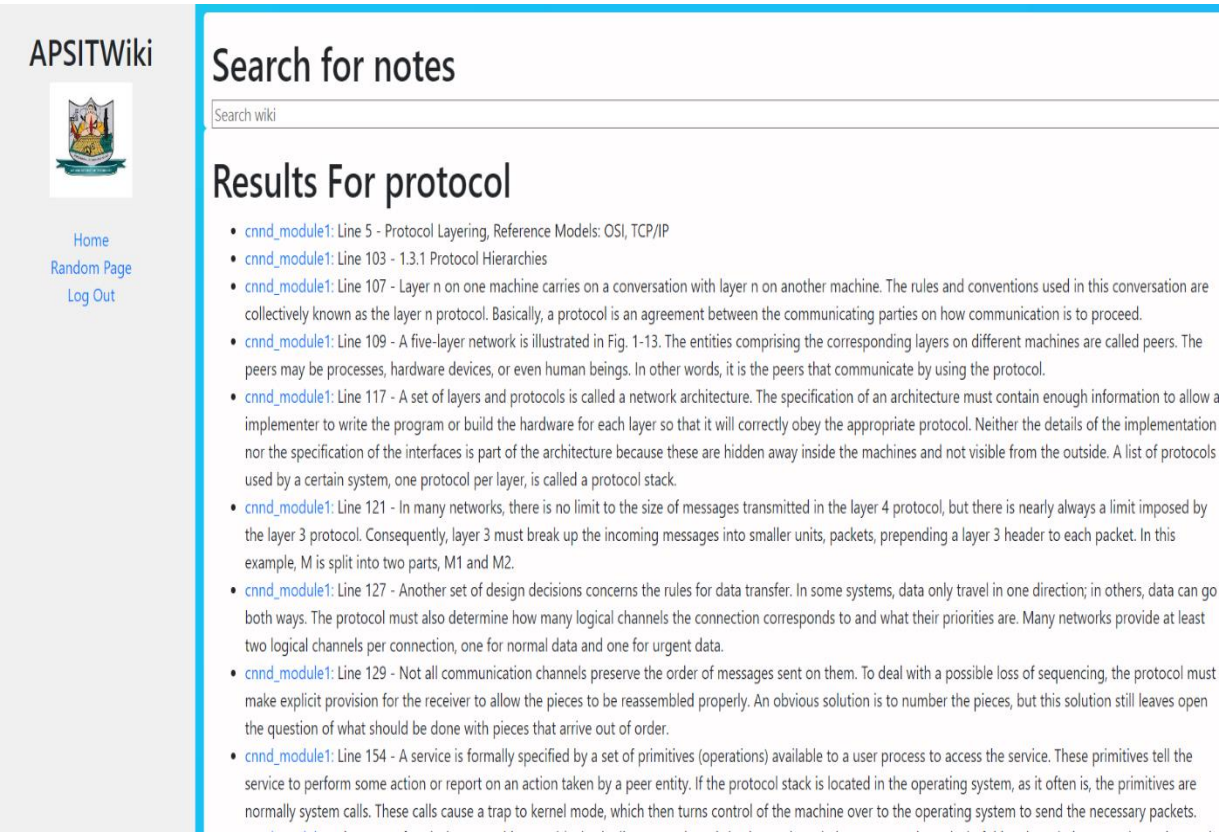
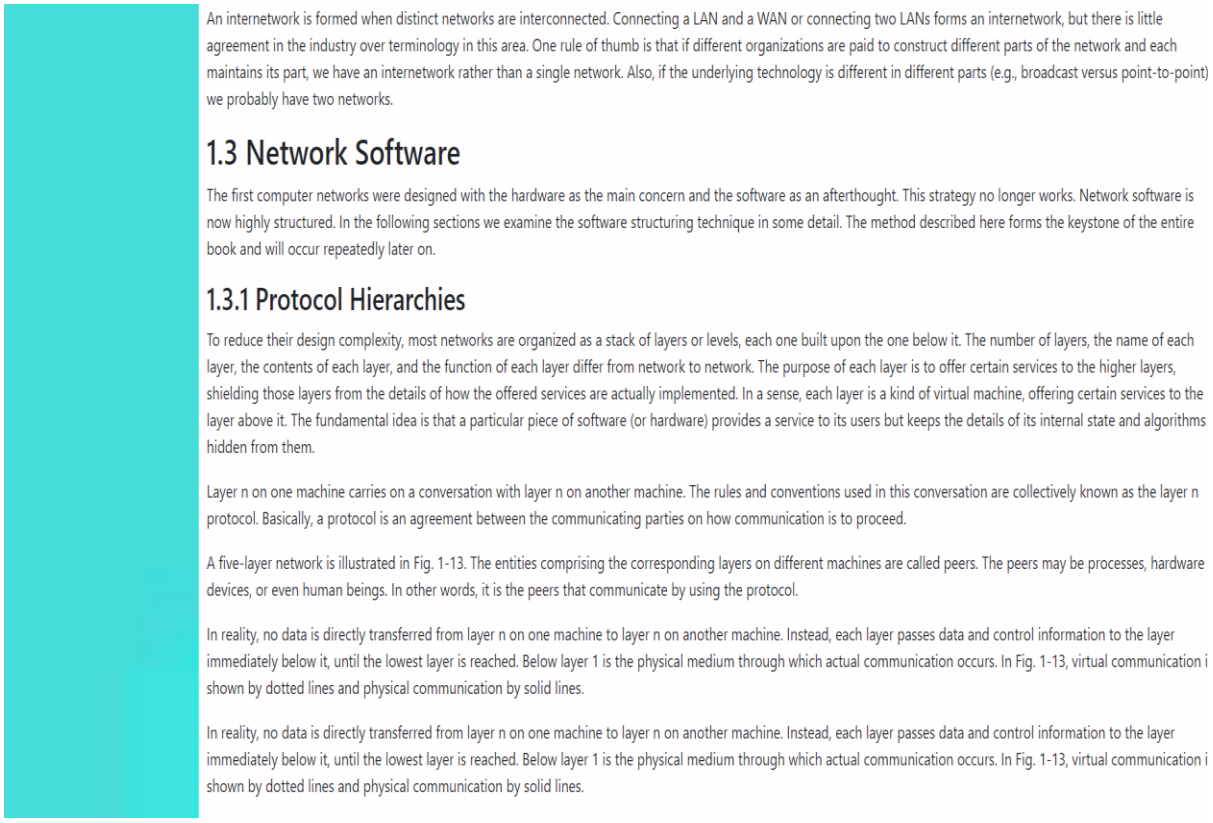


Fig 7.6: SEARCH RESULTS PAGE

- This is the Search Page
- Here the search results are shown.
- Each result contains a hyperlink that shows the module it belongs to.
- The result also shows the exact line of the occurence of the searched term within the module.



An internetwork is formed when distinct networks are interconnected. Connecting a LAN and a WAN or connecting two LANs forms an internetwork, but there is little agreement in the industry over terminology in this area. One rule of thumb is that if different organizations are paid to construct different parts of the network and each maintains its part, we have an internetwork rather than a single network. Also, if the underlying technology is different in different parts (e.g., broadcast versus point-to-point) we probably have two networks.

1.3 Network Software

The first computer networks were designed with the hardware as the main concern and the software as an afterthought. This strategy no longer works. Network software is now highly structured. In the following sections we examine the software structuring technique in some detail. The method described here forms the keystone of the entire book and will occur repeatedly later on.

1.3.1 Protocol Hierarchies

To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

Layer n on one machine carries on a conversation with layer n on another machine. The rules and conventions used in this conversation are collectively known as the layer n protocol. Basically, a protocol is an agreement between the communicating parties on how communication is to proceed.

A five-layer network is illustrated in Fig. 1-13. The entities comprising the corresponding layers on different machines are called peers. The peers may be processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol.

In reality, no data is directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. In Fig. 1-13, virtual communication i shown by dotted lines and physical communication by solid lines.

In reality, no data is directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. In Fig. 1-13, virtual communication i shown by dotted lines and physical communication by solid lines.

Fig 7.7: SEARCH RESULTS PAGE

- When clicking of the search results, it redirects the user to the heading of the searched term.
- The line is also put into the center of the screen.
- The user will get the most efficient way to find through the topic and get directly to the notes part in need.

Chapter 8

Conclusion

This project marks a successful culmination of our efforts to develop a comprehensive and user-friendly dashboard for AR navigation and campus information access. We've integrated key functionalities like real-time assistance, intuitive search capabilities, and a user-friendly interface, all aimed at enhancing the overall user experience. While the chatbot's current scope is limited to pre-defined responses, its future expansion holds immense potential to provide even more personalized and dynamic support. We believe this dashboard represents a significant step forward in leveraging technology to create a more connected and informed campus environment. With continued refinement and enhancements, we envision this platform evolving into an indispensable tool for navigating and engaging with campus life, further enriching the student experience.

Chapter 9

Future Scope

The Project has a promising future with several potential enhancements and expansions that could further enrich its capabilities and user experience.

- **Enhanced Chatbot Capabilities:** The chatbot could evolve beyond predefined responses by incorporating natural language processing (NLP) and machine learning. This would enable it to understand and respond to a broader range of user queries, offering more personalized and contextually relevant assistance.
- **Expanded Learning Platform:** The learning platform could be enriched with features like discussion forums, quizzes, and interactive modules, fostering a more engaging and collaborative learning environment.
- **Campus Resource Integration:** The system could become a central hub for accessing various campus resources, integrating with library catalogs, event calendars, and student services.
- **Improved Accessibility:** The system could be optimized for accessibility, ensuring it's usable by individuals with disabilities through features like screen reader compatibility and alternative text for images.
- **Data-Driven Optimization:** The system could collect and analyze user data to gain insights into usage patterns, enabling continuous improvement and personalization of the user experience.
- **Personalized Learning:** By leveraging user data and machine learning, the system could offer personalized learning paths, recommending courses and resources tailored to individual student interests and learning styles.
- **Gamification:** The introduction of game-like elements, such as badges, points, and leaderboards, could boost user engagement and motivation within the learning platform.
- **Social Learning:** Integrating social features like discussion forums and collaborative projects could foster peer-to-peer learning and knowledge sharing.
- **Offline Functionality:** The system could provide offline access to certain features, such as maps and course materials, enhancing usability in areas with limited or no internet connectivity.
- **Wearable Device Integration:** The AR navigation experience could be extended to wearable devices like smart glasses, offering a more seamless and hands-free way to navigate the campus.
- **Search Metrics:** Adding features to filter out modules of where the searched term occurs.
- **Community**: Opening the doors to comments and groups among users. This will have student groups share selected part of notes within themselves.

References:

[1] APSIT, APSIT moodle server, Thane: APSIT, 2016.

[2] Daniel Roy-Greenfeld and Audrey Roy-Greenfeld, "Django content Management," Django, NY, 2018.

[3] Adithela, Stephen Paul, et al. , ""Django content management system evaluation and integration" Proceedings of the Practice and Experience on Advanced Research Computing.," ACM, NJ, 2018.

[4]

[5] Fabrizio Romano, Gaston C. Hillar, Arun Ravindran, Learn Web Development with Python, LA: Packt Publishing Ltd, 2018.

[6] X. Qiao, P. Ren, S. Dustdar, L. Liu, H. Ma and J. Chen, Web AR: A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights, DN: Proceedings of the IEEE, vol. 107, no. 4, pp. 651-666, 2019.

[7] F. L. M. K. K. P. N. A. M. B. A. D. M. K. a. E. T. Kostas Cheliotis, "A systematic review of application development in augmented reality navigation research," ResearchGate, SE, 2023.

[8] P. A. E. a. P. K. I. Satya Kiranmai Tadepalli, "Indoor Navigation Using Augmented Reality," ResearchGate, HYD, 2021.