Office Of Academic -Jain University
KMA and Jain event

in partial fulfillment for the award of the degree
of

MASTER OF SCIENCE IN COMPUTER SCIENCE
AND
INFORMATION TECHNOLOGY

Submitted by

Santosh KS

USN No: 23MSRCI060

Under the Guidance of

Dr.Ganesh D

Research & Project Coordinator – MSc(CSIT)

JAIN (Deemed to Be University)

JGi JAIN DEEMED-TO-BE UNIVERSITY
School Of
Computer
Science and IT

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

JAIN KNOWLEDGE CAMPUS
JAYANAGAR 9TH BLOCK
BANGALORE - 560069

JGi JAIN DEEMED-TO-BE UNIVERSITY

School Of
Computer
Science and IT

DEPARTMENT OF COMPUTER SCIENCE

AND INFORMATION TECHNOLOGY

Jain Knowledge Campus
Jayanagar 9th Block Bangalore, 560069

This is to certify that the project entitled

Office Of Academic -Jain University KMA and Jain event

is the bonafide record of project work done by

Santosh KS

USN No: 23MSRCI060
MSc-CSIT during the year
2023 - 2025

Dr.Ganesh D
Guide/Mentor
JAIN (Deemed to Be University)

Prof. Raghavendra R
Program Co-ordinator – MSc(CSIT)
JAIN (Deemed to Be University)

Dr. Suneetha K
Head, Department of Computer Science & IT
JAIN (Deemed to Be University)

JAIN DEEMED-TO-BE UNIVERSITY | School Of Computer Science and IT

CERTIFICATE

This is to certify that Santosh KS, USN No: 23MSRCI060 for the course of MSc-CSIT in the Department of CSIT, School of Computer Science and IT has fulfilled the requirements prescribed for the MSc-CSIT degree of the of JAIN (Deemed to be University).

The Project entitled, "OFFICE OF ACADEMIC JAIN UNIVERSITY - KMA AND JAIN EVENTS " was carried out under my direct supervision. No part of the dissertation was submitted for the award of any degree or diploma prior to this date.

Dr.Ganesh D
Guide/Mentor
JAIN (Deemed to Be University)

Prof. Raghavendra R
Program Co-ordinator – MSc(CSIT)
JAIN (Deemed to Be University)

Dr. Suneetha K
Head, Department of Computer Science & IT
JAIN (Deemed to Be University)

Name of the Examiner	**Signature with Date**
1.	....
2.	....

JGi JAIN DEEMED-TO-BE UNIVERSITY

School Of
Computer
Science and IT

DECLARATION

I affirm that the project work titled “OFFICE OF ACADEMIC JAIN UNIVERSITY - KMA AND JAIN EVENTS ”, being submitted in partial fulfillment for the award of MASTER OF SCIENCE IN COMPUTER SCIENCE AND INFORMATION TECHNOLOGY is the original work carried out by me. It has not formed part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

Santosh KS
USN Number: 23MSRCI060

JGi JAIN DEEMED-TO-BE UNIVERSITY | School Of Computer Science and IT

ACKNOWLEDGEMENT

I would like to acknowledge the following people, who have encouraged, guided and helped to accomplish my report to award my degree at The JAIN (Deemed to be University), Department of Computer Science Information Technology, School of Computer Science and IT:

1. Thesis advisor and mentor Prof Raghavendra R for guiding me through pivotal moments of my study and professional career and for always being there to make sure that my progress was reviewed, documented and acknowledged. His/Her encouragement has been the greatest source of inspiration and confidence for me as a designer and artist.

2. Faculty and staff members of **Department of Computer Science & IT** for sharing their expertise and for always showing their interests in my work.

3. I also would like to extend my thanks to my friends, and particularly to my guide , for his efforts to make my dissertation/ report a more effective.

4. Finally, I would like to thank my family, to whom this work is dedicated, for their support and encouragement during these years.

Special Thanks to:

- Dr. SUNEETHA K, Head, Department of Computer Science and IT, JAIN (Deemed-to-be University)
- Prof. RAGHAVENDRA R, Programme Coordinator, MSc-CSIT Programme, Department of Computer Science and IT JAIN (Deemed-to-be University)
- Dr. GANESH D, Research & Project Coordinator, Department of Computer Science and IT, JAIN (Deemed-to-be University)
- Prof. HARIPRIYA V, TDPCL Coordinator, Department of Computer Science and IT, JAIN (Deemed-to-be University)

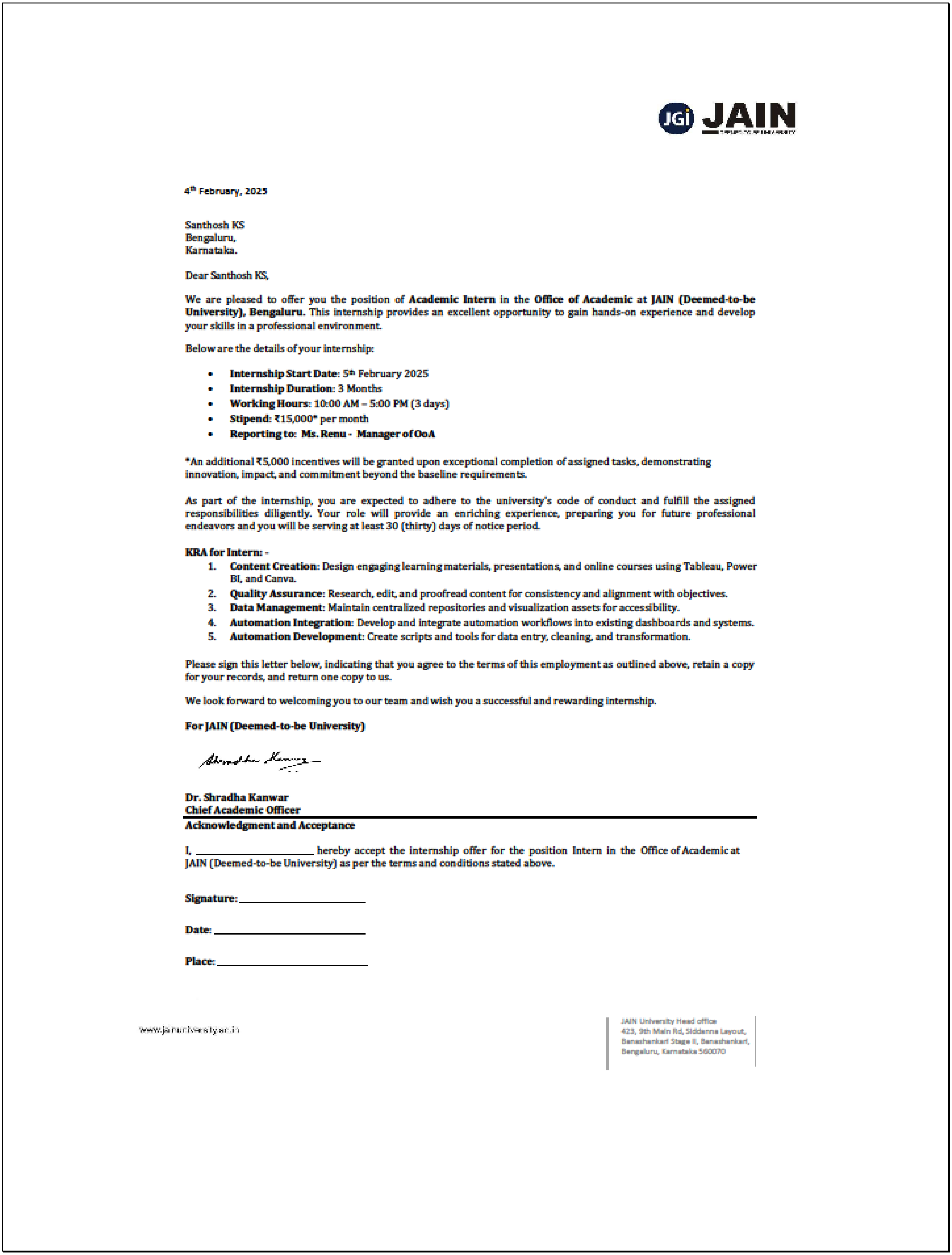
INTERNSHIP REPORT ON

Office Of Academic -Jain University KMA and Jain event

Santosh KS (23MSRCI060)

Internship Duration : 05/02/2025 – 30/06/2025
Organization Name : Office Of Aademic – Jain(Deemed-to-be) University

Manager Name : Ms. Renu
Position : Full stack Web Developer Intern
University : Jain(Deemed-to-be) University



4th February, 2025

Santhosh KS
Bengaluru,
Karnataka.

Dear Santhosh KS,

We are pleased to offer you the position of **Academic Intern** in the **Office of Academic** at **JAIN (Deemed-to-be University), Bengaluru.** This internship provides an excellent opportunity to gain hands-on experience and develop your skills in a professional environment.

Below are the details of your internship:

- **Internship Start Date**: 5th February 2025
- **Internship Duration**: 3 Months
- **Working Hours**: 10:00 AM – 5:00 PM (3 days)
- **Stipend**: ₹15,000* per month
- **Reporting to: Ms. Renu - Manager of OoA**

*An additional ₹5,000 incentives will be granted upon exceptional completion of assigned tasks, demonstrating innovation, impact, and commitment beyond the baseline requirements.

As part of the internship, you are expected to adhere to the university's code of conduct and fulfill the assigned responsibilities diligently. Your role will provide an enriching experience, preparing you for future professional endeavors and you will be serving at least 30 (thirty) days of notice period.

KRA for Intern: -

1. **Content Creation**: Design engaging learning materials, presentations, and online courses using Tableau, Power BI, and Canva.
2. **Quality Assurance**: Research, edit, and proofread content for consistency and alignment with objectives.
3. **Data Management**: Maintain centralized repositories and visualization assets for accessibility.
4. **Automation Integration**: Develop and integrate automation workflows into existing dashboards and systems.
5. **Automation Development**: Create scripts and tools for data entry, cleaning, and transformation.

Please sign this letter below, indicating that you agree to the terms of this employment as outlined above, retain a copy for your records, and return one copy to us.

We look forward to welcoming you to our team and wish you a successful and rewarding internship.

For JAIN (Deemed-to-be University)

Shradha Kanwar

Dr. Shradha Kanwar
Chief Academic Officer

Acknowledgment and Acceptance

I, ______________________ hereby accept the internship offer for the position Intern in the Office of Academic at JAIN (Deemed-to-be University) as per the terms and conditions stated above.

Signature: ________________________

Date: ____________________________

Place: ____________________________

www.jainuniversity.ac.in

JAIN University Head office
423, 9th Main Rd, Siddanna Layout,
Banashankari Stage II, Banashankari,
Bengaluru, Karnataka 560070

ABSTRACT

Jain (Deemed-to-be) University is a prominent institution known for its commitment to academic excellence, innovation, and research-driven education. The university offers a wide spectrum of undergraduate, postgraduate, and doctoral programs across various disciplines. To support the continuous improvement of academic operations and digital infrastructure, the **Office of Academic Affairs** plays a vital role in shaping institutional academic strategies, curriculum development, and policy planning.

As an **Academic Intern** in the Office of Academic Affairs, I had the opportunity to contribute to the university's ongoing digital initiatives by developing web-based platforms and tools for various **schools and departments**. My work primarily focused on designing, building, and deploying custom websites and data management systems that cater to the specific needs of academic units. These tools are integrated under the broader framework of the **Knowledge Management Architecture (KMA)**—a centralized digital system aimed at improving the accessibility and efficiency of academic data across the institution.

Throughout the internship, I was responsible for end-to-end web development tasks including frontend design, backend logic, database integration, and deployment. I used technologies such as **HTML, CSS, JavaScript, Python (Django framework)**, and **cloud platforms like AWS**

to ensure scalable and secure access to academic resources. The project enabled each department to manage their academic documents, events, syllabi, faculty information, and institutional outcomes more efficiently through a unified interface.

Overall, this internship experience provided hands-on exposure to full-stack development in an academic setting and contributed directly to the digital transformation of Jain (Deemed-to-be) University's academic ecosystem.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND OF THE ORGANIZATION

Jain University is a prestigious institution committed to academic excellence, research, and innovation. It offers a wide spectrum of Undergraduate, Postgraduate, and Doctoral programs across disciplines such as Engineering, Management, Sciences, Humanities, and more. With its strong industry collaborations, global partnerships, and focus on outcome-based education, the university provides students with opportunities to engage in real-world challenges and research driven learning.

The **Office of Academic Affairs** plays a pivotal role in steering the academic direction of the university. It is responsible for academic planning, curriculum design and review, policy formulation, regulatory compliance, academic scheduling, and coordination among departments. The office acts as a bridge between faculty, students, and administrative units to ensure smooth academic operations and quality assurance across all programs.

As an **Academic Intern** in this department, I have been entrusted with a critical project

focused on enhancing data accessibility and management through the **Knowledge Management Architecture (KMA)** system. The KMA system is envisioned as a centralized digital platform for organizing and accessing academic records, calendars, circulars, reports, and event documentation. It supports easy retrieval, secure access, and streamlined dissemination of academic information to relevant stakeholders including faculty, administrative staff, and students.

My primary role in this project involves assisting in the design, development, and implementation of backend functionalities that support one-click accessibility to academic data.By integrating the system with institutional workflows, the KMA enhances operational efficiency, reduces information silos, and supports informed decision-making. This project aligns with the university’s mission of adopting technology-driven solutions to foster academic excellence and institutional growth.

This internship has enriched my understanding of academic administration and information systems. It has also given me hands-on experience with real-time problem solving, collaborative development, and the implementation of digital solutions in higher education.

.

1.2 OVERVIEW OF THE INTERNSHIP PROGRAM

The Web Developer Internship at **Jain University** offered me an excellent opportunity to applymy technical knowledge in a professional and academic environment. Conducted under the **Office of Academic Affairs**, the internship was part of an initiative to improve data management and accessibility across the institution through the development of the

Knowledge Management Architecture (KMA) system.

At the beginning of the internship, I set the following goals:

- To gain hands-on experience in **web development** using **Python** and the **Django framework**.
- To contribute actively to the **Knowledge Management Architecture (KMA)** project by developing features focused on **data accessibility** and **user-friendly interfaces**.
- To integrate **Amazon Web Services (AWS)** for secure and efficient **cloud-based deployment** of the system.
- To enhance my ability to work effectively in a **professional team environment**, adopting **industry best practices** in software development.

Internship Highlights

Throughout the internship:

- I worked on backend development using Django, creating APIs and integrating them with the user interface.
- I implemented modules that allow streamlined uploading, viewing, and management of academic records and events.
- I participated in team meetings, reviews, and collaborative development workflows, using tools like Git, Docker, and cloud-based deployment on AWS.
- I gained exposure to the operational and technical processes involved in academic administration systems.

This internship has significantly enhanced both my **technical proficiency** and **professional communication skills**. It also provided a deeper understanding of how digital solutions can support large-scale academic processes.

Overall, this experience was invaluable in bridging the gap between academic learning and practical implementation, preparing me for real-world challenges in the field of **web development and cloud computing**.

1.3 OBJCTIVES OF THE INTERNSHIP

The primary objective of the internship was to bridge the gap between academic learning and real-world professional experience. The internship at **Jain University**, under the guidance of the **Office of Academic Affairs**, aimed to provide practical exposure to web development, software implementation, and institutional digitization processes. This opportunity allowed me to work in a structured environment where I could apply theoretical knowledge and develop technical, analytical, and soft skills essential for a successful career in software and web development.

The following were the core objectives of the internship:

1. **Hands-on Experience in Web Development**
   To gain practical exposure in designing and developing dynamic web applications using **Python** and the **Django framework**, including the implementation of models, views, templates, and database integration.
2. **Contribution to the Knowledge Management Architecture (KMA)**
   To contribute to the development of the **Knowledge Management Architecture** system,a centralized platform intended to enhance data accessibility and streamline academic record management across the institution.
3. **Integration of Cloud Technologies (AWS)**
   To learn and implement **cloud-based deployment** using **Amazon Web Services (AWS)**ensuring scalability, data security, and remote accessibility of the academic web application.
4. **Understanding Academic Workflow Automation**
   To observe and understand how academic processes and workflows operate in a university setting, and how they can be digitized for efficiency, reliability, and transparency.
5. **Improvement of Technical and Soft Skills**
   To enhance personal and professional development by working collaboratively in a
6. team, participating in code reviews, applying **version control systems** (e.g., Git), and following **industry-standard development practices**.
7. **Problem Solving and Innovation**
   To identify challenges in academic data management and propose innovative, user-friendly solutions through well-structured backend systems and interfaces.

Through this internship, I aimed to strengthen my foundation in full-stack development and understand how digital solutions can support institutional goals in higher education. The experience also equipped me with the discipline and adaptability required to succeed in futuresoftware development roles.

CHAPTER 2
PROJECT/TASK DESCRIPTION

2.1 PROJECT TITLE

Development of Knowledge Management Architecture (KMA) and Jain Event Management System

The title of the project undertaken during the internship is *"Development of Knowledge Management Architecture (KMA) and Jain Event Management System."* This project was conceptualized and implemented under the supervision of the **Office of Academic Affairs** at **Jain University**, with the primary aim of creating a centralized platform to manage academic data and events more effectively.

The project combines two core components:

1. **Knowledge Management Architecture (KMA)** – A backend-driven digital system to handle academic records, circulars, reports, syllabi, calendars, and other institutional documentation in a secure and accessible way.
2. **Jain Event Management System** – A web-based platform that allows departments and administrative units to create, publish, and monitor university events in a structured and collaborative environment.

The integration of both systems into a unified platform not only enhances data accessibility but also reduces administrative burden, ensures better communication across departments, and promotes a paperless workflow in line with institutional digital transformation goals.

Jain University is committed to integrating technology with education to ensure smooth academic operations and enhanced data-driven decision-making. In this context, the internship project was designed to support the university's vision of digitizing institutional workflows and improving the efficiency of academic and event-related data management.

The **Knowledge Management Architecture (KMA)** system addresses the challenge of decentralized and scattered academic data. Previously, academic records, schedules, and notices were managed manually or stored across multiple platforms, which often led to delays in retrieval, loss of data, and inefficiencies. The KMA system solves this by acting as a centralized knowledge hub where all relevant academic content can be uploaded, categorized, and accessed based on user roles such as administrators, faculty, or students. It supports uploading of various document types (PDFs, Word files, images) and includes features like tagging, version control, search functionality, and data security through role-based access control.

The **Jain Event Management System**, developed as part of the same platform, focuses on managing academic and non-academic events organized within the university. This includes seminars, workshops, student activities, conferences, training sessions, and other institutional programs. The system provides modules for event creation, scheduling, automated notifications, and archiving of past events for future reference. It streamlines the event coordination process, avoids scheduling conflicts, and enables better communication between organizers and participants.

Throughout the internship, I was responsible for implementing both backend and frontend components of the system using **Python**, the **Django framework**, and **JavaScript** for interactivity. I also worked on integrating the application with **Amazon Web Services (AWS)** to enable **cloud-based deployment**, ensuring data availability, reliability, and scalability. The use of AWS also introduced secure authentication, storage (via S3 buckets), and potential integration with other cloud services in the future.

Working on this project helped me explore real-world applications of my technical skills in web development. I gained valuable experience in:

- Understanding the lifecycle of academic data and event management
- Designing efficient data models and database schemas
- Creating reusable components and templates for administrative dashboards
- Implementing role-based authentication and user authorization
- Applying cloud computing practices for real-world deployment

Moreover, the project also allowed me to collaborate with academic administrators, IT professionals, and faculty members, improving my soft skills and professional communication. It provided an ideal platform to learn about the challenges faced in higher education administration and how technical solutions can make meaningful contributions.

In conclusion, the KMA and Jain Event Management project not only gave me the opportunity to contribute to the university's digital initiatives but also enhanced my understanding of how software solutions can transform institutional processes. It was a rewarding experience that aligned both my academic learning and future career aspirations in the field of full-stack development and digital transformation in education.

2.2 PROBLEM STATEMENT

One of the most immediate and recurring technical challenges was managing bugs, especially during **database migrations** and **API integrations**. As the project involved continuous enhancements and model changes, applying migrations frequently led to errors such as mismatched fields, missing dependencies, and inconsistent table states. At times, these issues even caused the development server to crash or behave unpredictably.

For example, while updating the Event model to include additional metadata fields like "Department Name" and "Approval Status", I faced multiple migration conflicts. These required resetting the migration history or manually editing migration files—an advanced task that demanded a deep understanding of Django's ORM and migration system.

API integration also presented difficulties. Building API endpoints to connect the Django backend with the frontend involved ensuring proper request/response formatting, status code handling, and data serialization. A lack of validation logic initially resulted in broken API calls, unexpected behavior, and failed form submissions. These issues significantly delayed the development process until robust error-handling mechanisms were put in place.

As the system scaled to accommodate more users and data entries, performance issues began to emerge. Initially, the system worked well with limited test data. However, as real-world academic records, events, and user roles were entered into the system, **page load times increased**, and several views began timing out or returning partial results.

To improve performance, I had to examine the efficiency of database queries, particularly in views with filtering or pagination. I optimized database access using Django's select_related() and prefetch_related() methods to reduce redundant queries. I also created indexes for frequently filtered fields like event_date, department, and category.

Additionally, caching techniques were considered to reduce redundant data loading, especially for frequently accessed data like the academic calendar or department-specific announcements. By introducing optimizations at the model, view, and template levels, performance was gradually improved to ensure faster load times and a smoother user experience.

brought in another layer of complexity. Setting up the **EC2 instance**, configuring **security groups**, handling **gunicorn/nginx** for serving the Django app, and ensuring **SSL configuration** for secure access posed multiple learning challenges. For a developer with limited prior cloud experience, this required substantial research, experimentation, and error resolution. Challenges included ensuring proper port access, environment variable management, database connectivity, and persistent storage on cloud volumes.

the project—coding, documentation, meetings, bug fixes, learning new tools—proved to be one of the most significant non-technical challenges. With tight weekly milestones and team updates, maintaining consistent progress while also learning on the job demanded effective **prioritization and scheduling**.

In the beginning, I found myself spending excessive time perfecting minor features while more critical tasks remained pending. I had to restructure my workflow by using a **task matrix** that classified tasks based on urgency and importance, enabling better focus and timely completion.

At times, project requirements were not fully defined or underwent changes due to new institutional needs. For instance, an event tagging feature was added mid-sprint after feedback from faculty. This led to adjustments in both frontend forms and backend models, often requiring rollback and refactoring of previously completed modules.

Additionally, there was a communication gap between developers and non-technical stakeholders. Explaining the limitations and timelines of technical implementations in non-technical terms required improvement in communication and presentation skills. To overcome this, I started attending more regular briefings and maintained a feature log for stakeholder review.

Maintaining proper and consistent documentation of the codebase and deployment procedures was initially neglected due to time pressure. However, as the project grew in complexity, I realized the necessity of documenting setup instructions, environment variables, model relationships, and API endpoints to ensure maintainability.

Version control through Git also posed early challenges. Merge conflicts and incorrect branching sometimes caused code loss or inconsistency. I eventually adopted **branch naming conventions**, **frequent commits**, and **pull request reviews** to streamline collaboration and ensure code safety.

- **Debugging Tools and Techniques**: Used Django Debug Toolbar, custom loggers, and manual testing to locate and resolve bugs efficiently.
- **API Testing Tools**: Tools like Postman and Swagger were used to test and document API endpoints, ensuring smooth integration between frontend and backend components.
- **Optimized Query Writing**: Reviewed all ORM queries and applied optimization techniques like lazy loading and indexing to enhance performance.
- **Cloud Deployment Assistance**: Followed AWS deployment documentation, online tutorials, and support from mentors to configure EC2, setup HTTPS, and deploy the Django project securely.
- **Time Management Strategies**: Introduced weekly planning using Trello and Google Sheets to track task completion and reduce time waste.
- **Clear Documentation**: Created detailed README files, API documentation, and deployment instructions to improve project handover and future scalability.
- **Improved Communication**: Participated in stand-up meetings, stakeholder discussions, and project briefings to ensure alignment and avoid last-minute requirement changes.

Overcoming these technical and managerial problems allowed me to grow significantly as a developer and team member. I learned that software development in a real-world setting is about more than just writing code—it's about writing **efficient**, **scalable**, and **maintainable** code that meets organizational objectives. I also learned to **listen actively**, **collaborate cross-functionally**, and **communicate effectively**, especially when working on multi-stakeholder projects like KMA.

These challenges shaped my ability to think critically, manage pressure, adapt quickly, and solve problems creatively—skills that are essential for any software engineering role.

2.3 OBJECTIVES OF THE PROJECT/TASK

The primary objective of the project undertaken during my internship was to support the Office of Academic Affairs at Jain University by designing and developing a centralized digital platform that enhances data accessibility, process transparency, and operational efficiency. The two major modules involved in this task were:

The **Knowledge Management Architecture (KMA)** System: A framework to digitally organize and provide controlled access to academic documents, institutional reports, and departmental data.

The **Jain Event Management System**: A platform to streamline the creation, management, tracking, and reporting of university-wide events.

The unified goal was to enhance institutional digital infrastructure while providing an intuitive user experience for both administrators and faculty members. Below is a detailed elaboration of the objectives that guided the development of this project.

The project aimed to utilize the Django web framework, known for its rapid development capabilities, scalability, and built-in security features. Django's robust architecture allowed for clean model-view-template (MVT) separation, which ensured modularity and reusability of components.

Key features developed under this objective included:

- User Authentication and Authorization (Admin, Faculty, Staff)
- Dynamic Form Handling and Submission
- Database Model Creation and Migration
- CRUD (Create, Read, Update, Delete) Functionalities for Events, Users, and Academic Records

To ensure the front end of the application remained dynamic and responsive, REST APIs were created using Django REST Framework. These APIs enabled frontend clients—both browser-based and mobile—to interact with the backend seamlessly.

Objectives for API development:

Ensure secure access using JWT (JSON Web Tokens)

Validate user input at the API level

Support pagination, filtering, and search

Provide JSON responses for frontend rendering in React (planned in later stages)

A well-structured database schema was one of the cornerstones of the KMA system. The objective here was to:

Create normalized models for academic departments, specializations, semesters, modules, events, and user roles

Implement foreign key relationships for proper data linking

Allow version control for uploaded files such as syllabi, results, or course matrices

Minimize data redundancy and ensure referential integrity

An essential component of the project was to expose the application to real users via a cloud-based deployment. AWS EC2 was chosen due to its flexibility and affordability. Key deployment goals included:

- Configuring a secure EC2 instance using Ubuntu
- Installing and configuring production-grade servers (Gunicorn + Nginx)
- Setting up environment variables and secret key management
- Ensuring HTTPS through SSL certification using Let's Encrypt

Implement Access Control and Data Protection

Given the sensitive nature of academic data, it was necessary to enforce strict access control mechanisms. The Django admin interface was customized to:

Grant role-based access (e.g., only HODs can approve events, only Admins can manage users).Log user activity and manage permission escalation.Prevent unauthorized access to confidential academic documents.Additionally, file uploads were stored securely and served only to authenticated users. Security headers and input sanitization measures were put in place to prevent injection attacks and CSRF vulnerabilities.

Streamline Academic Event Management

The previous process for managing academic events relied heavily on email communications and spreadsheets. The objective of this system was to:

Enable HODs and faculty to propose events using an online form

Allow the Academic Office to approve, reject, or request edits

Automatically track event status (pending, approved, completed)

Generate monthly reports for documentation and analysis

The goal was to replace redundant manual processes with a systematic, paperless workflow.

Digitize the Knowledge Repository

KMA was envisioned as a digital repository for institutional knowledge. The objectives here were.Provide structured document upload functionality for syllabi, TLEPs, results, and course matrices.Allow students, faculty, and stakeholders to view department-specific data based on permissions.Support document versioning and categorization.Enhance searchability and reduce dependency on physical archives

Design a Simple, Responsive, and Intuitive Interface

- Although the initial version relied on Django templates, the plan included integrating React.js for an interactive frontend. The design objectives were:
- Keep navigation minimal with clear sectioning (Home, Events, Admin Panel, KMA)
- Ensure mobile responsiveness and cross-browser compatibility
- Provide tooltips, status indicators, and feedback messages to improve user interaction
- Reduce the learning curve for non-technical users such as professors or department staff

Accessibility and Inclusivity

The goal was to ensure the platform could be used by people with varying levels of digital literacy. Accessibility considerations included:

- High-contrast color schemes

- Keyboard navigability
- Simple error messages and help texts
- Support for future language localization (if needed)

Apply Academic Knowledge to Real-World Scenarios

This project allowed me to bridge the gap between theoretical concepts learned during coursework and real-life application. Objectives included:

- Understand software development lifecycle phases: planning, development, testing, deployment
- Apply database design principles practically
- Follow software engineering best practices such as version control, modularity, and testing

Develop Collaboration and Communication Skills

A key learning goal was to function effectively in a team environment. I participated in:

- Regular review meetings
- Task assignment and sprint planning sessions
- Demonstrations and feedback discussions with stakeholders

These helped improve my ability to communicate technical ideas clearly and work toward shared goals.

.

2.4 SCOPE OF WORK

The initial scope involved designing a scalable and modular architecture that could support multiple modules in a single unified platform. This included:

- **Designing Models and Schema:** Database schema development for users, departments, events, academic resources (syllabi, results, course matrix, etc.), permissions, and logs.
- **Modular System Structure:** Creating reusable components for handling user access, uploading documents, managing event status, and rendering dynamic templates.
- **User Role Hierarchy:** Implementing distinct access levels for Admin, HODs, Faculty, and Event Coordinators.

The architecture was planned with future scalability in mind—ensuring that more departments and features could be added later without structural overhauls.

Backend Development Scope

The core backend development was performed using the Django framework in Python. The scope in this domain included:

Authentication System: Creating login and logout features, password protection, and secure access control for various users.

Admin Panel Customization: Extending Django's admin interface to allow academic office staff to add departments, specializations, events, users, and academic documents.

REST API Integration (Planned): Partially developed APIs to allow frontend consumption (especially for the upcoming React integration).

Form Handling and Validation: Handling complex form submissions for event proposals and academic document uploads with validations.

Search and Filter Functionality: Creating filters based on department, semester, and academic year to locate documents and events efficiently.

Frontend and User Interface Scope

Although the primary focus was backend development, frontend components were also developed using Django templating. The frontend scope included:

User Dashboard Creation: Dashboard views for admins and faculty members to monitor events and document statuses.

Interactive Forms: HTML templates with Bootstrap for submitting new events, uploading files, and managing modules.

Navigation System: Designing intuitive navigation for viewing academic content by year, department, and course.

Responsive Design (Partially Implemented): Ensuring that the system is mobile-friendly and works across various screen sizes.

For the second phase of the internship, the plan is to replace the templated front end with a modern React.js interface for more dynamic interactions.

Database and File Management Scope

A major component of the scope involved designing and managing a PostgreSQL database linked to Django's ORM. Work done in this area included:

- **Database Configuration and Optimization**
  - Creating migrations for models
  - Optimizing foreign key relationships
  - Reducing redundancy and ensuring data normalization
- **File Upload Management**
  - Supporting the upload of PDF documents including syllabi, course matrices, and results
  - Organizing files in department-wise and semester-wise directories
  - Restricting access to sensitive files based on user roles
- **Version Control for Documents**

Cloud Deployment Scope

To make the application accessible institution-wide, cloud deployment was necessary. This part of the work involved:

- **AWS EC2 Configuration**
  - Launching a secure Ubuntu instance
  - Installing and configuring required packages (Python, Gunicorn, Nginx, PostgreSQL)
  - Setting up environment variables and static file handling
- **Domain Mapping and SSL Integration**
  - Mapping a subdomain to the server
  - Installing SSL certificates to enable HTTPS
- **Maintenance Tasks**
  - Restarting services on failure
  - Monitoring logs
  - Updating application as needed

This ensured the application was available 24/7 for usage by authorized stakeholders.

Project Management and Collaboration Scope

The scope was not limited to technical implementation—it also extended into communication and project coordination. Some key areas included:

- **Requirement Analysis**
    - Participating in discussions with department heads to understand document workflows and event planning challenges
    - Gathering and interpreting requirements for features

Documentation

- Writing internal documentation for system architecture and model relationships
- Preparing user manuals and SOPs for academic staff

Task Prioritization and Sprint Planning

- Using task boards to manage day-to-day activities
- Prioritizing tasks based on urgency and importance

Reporting

- Providing weekly updates on development progress
- Participating in code reviews and testing phases

Limitations and Out-of-Scope Items

While a broad range of work was completed during the internship period, some components were identified as out of scope for the initial phase due to time or resource constraints:

- **Full Frontend React Implementation:** Postponed to the next half of the internship.
- **Mobile App Interface:** Although planned as a long-term goal, mobile-specific development was not part of the current scope.
- **Integration with ERP or LMS:** Direct integration with university ERP or learning management systems is under future consideration.
- **Advanced Analytics and Reporting:** Basic event and document reports were implemented, but deep analytics or dashboards using charts (e.g., Chart.js or D3.js) are future enhancements.

The scope of work covered during the internship encompassed comprehensive responsibilities across backend development, database management, admin tools, cloud deployment, and documentation. The work significantly contributes to Jain University's goal of digitizing academic workflows and enhancing operational transparency. Each module—whether it was user authentication, file uploads, or event management—was carefully designed to align with real-world institutional needs.

CHAPTER 3
TECHNOLOGIES AND TOOLS USED

3.1 PROGRAMMING LANGUAGES

During the course of my internship at **Jain University** as a Web Developer Intern, I extensively worked with a set of modern and industry-relevant programming languages. These languages played a vital role in developing and implementing various components of the **Knowledge Management Architecture (KMA)** system and the **Jain Event Portal**. The primary languages used included **Python**, **JavaScript**, **HTML**, and **CSS**. Each language served a distinct purpose in the backend, frontend, and user interface development.

1. Python

Python was the **core backend programming language** used throughout the development of the project. Its simplicity, readability, and robust library support made it ideal for building scalable and efficient web applications.

Framework: Django (Python Web Framework)

Responsibilities handled with Python:

- Backend logic for user authentication, permissions, and role management
- Database model creation and schema migrations
- Handling business logic for academic events and file uploads
- Admin panel customization for event and document management

Advantages:

- Quick development cycles
- Vast open-source libraries
- Seamless integration with PostgreSQL and AWS

JavaScript

JavaScript was primarily used to add interactivity and enhance user experience in the frontend part of the application. Though the core frontend React.js implementation is scheduled for the second phase, JavaScript helped with client-side validations and dynamic form behavior.

Responsibilities handled with JavaScript:

- Enhancing UI/UX by validating form inputs before submission
- Toggling between event lists and academic year tabs
- Preparing groundwork for future React.js integration

Advantages:

- Dynamic behavior without full-page reloads
- Strong compatibility with Django templates
- Paves the way for Single Page Application (SPA) development in the future

HTML (HyperText Markup Language)

HTML was used to structure the web pages, form the skeleton of the user interface, and define content hierarchy. It was integrated with Django templates to deliver dynamic content rendered from the backend.

Usage examples:

- Designing login and dashboard pages
- Structuring forms for event submission and document uploads
- Creating reusable UI blocks for academic content

CSS (Cascading Style Sheets)

CSS was used for styling the user interface and ensuring a visually appealing and consistent design across pages. Although Bootstrap was used for faster styling, custom CSS played a role in overriding default themes and ensuring responsiveness.

- **Usage examples:**
  - Styling buttons, input fields, and navigation menus
  - Making the design mobile-friendly
  - Ensuring visual distinction between user roles and permissions

The selection and effective usage of programming languages such as **Python, JavaScript, HTML, and CSS** played a pivotal role in the successful development of the KMA system and Event Management Platform. These languages enabled the construction of a well-structured, dynamic, and scalable application that aligns with modern software engineering practices. The combination of Python's robust backend capabilities with JavaScript's interactivity and HTML/CSS's structural and visual strength resulted in a full-stack solution that serves the university's academic needs efficiently.

3.2 SOFTWARE/FRAMEWORKS

As part of the internship at **Jain University**, I utilized a range of powerful **software tools and frameworks** to design, develop, and deploy a robust web application. These frameworks enhanced productivity, streamlined development, and ensured scalability and maintainability of the system. The primary frameworks and tools used were **Django**, **Bootstrap**, **PostgreSQL**, and **AWS Cloud Services**.

Django (Python Web Framework)

Django was the backbone of the entire backend development process. It is a high-level Python web framework that encourages rapid development and clean, pragmatic design.

- **Features leveraged:**
    - **Model-View-Template (MVT) architecture** for clean separation of concerns
    - **Built-in authentication system** for user registration, login, and permission control
    - **Admin panel customization** to manage departments, users, events, and academic documents
    - **ORM (Object-Relational Mapping)** for seamless database interactions
- **Why Django?**
    - Highly secure and prevents SQL injection, XSS, and CSRF attacks
    - Scales well for future integration of additional academic modules
    - Great community support and extensive documentation

Bootstrap (Frontend CSS Framework)

Bootstrap was used to design a responsive and mobile-friendly frontend interface. It helped create visually consistent UI components with minimal effort.

- **Components used:**
    - Forms, buttons, cards, modals, alerts, and navigation bars
    - Grid layout system to maintain consistency across screen sizes

Advantages:

- Rapid prototyping of user interfaces
- Integration with custom CSS for flexibility
- Responsive by default – works on desktop, tablet, and mobile views

(Relational Database Management System)

.

- **Database Responsibilities:**
  - Storing user profiles, departmental data, events, academic uploads (syllabus, results, etc.)
  - Maintaining referential integrity with foreign keys
  - Query optimization for performance in data retrieval

AWS (Amazon Web Services)

To make the application publicly accessible and secure, **AWS Cloud Platform** was used for deployment and hosting.

- **Services used:**
  - **EC2 (Elastic Compute Cloud):** Hosted the Django server
  - **S3 (Simple Storage Service):** Planned for document storage in scalable buckets
  - **Route 53:** For domain management and DNS routing
  - **SSL/TLS certificates:** Ensured secure HTTPS access
- **Benefits of using AWS:**
  - High availability and scalability
  - Industry-standard cloud security
  - Flexibility to integrate more services like RDS, CloudFront, or Lambda

Git and GitHub (Version Control System)

Version control was handled using **Git**, and the codebase was maintained on **GitHub**.

- **Features used:**
  - Branching and merging for feature-specific development
  - Commit history for tracking changes
  - Collaboration with supervisor and code reviews
- **Benefits:**

  - Ensured code integrity and rollback in case of errors
  - Enabled collaborative development with team members
  - Made the deployment pipeline more manageable

Visual Studio Code (Development Environment)

VS Code served as the primary Integrated Development Environment (IDE) during the internship.

- **Extensions used:**
  - Python support with linting and auto-formatting
  - Django snippets for faster coding
  - Git integration for easy commits and merges

3.3 TOOLS AND TECHNOLOGIES

Visual Studio Code (VS Code) was the primary development environment used throughout the internship.

Syntax highlighting and intelligent code completion (IntelliSense) for Python, JavaScript, HTML, and CSS.

- Debugging tools to identify and fix errors quickly.
- Git integration to manage version control from within the editor.
- Extensions such as Python support, Django snippets, and Prettier for code formatting.Lightweight and fast, enabling seamless development workflows.
- Customizable interface and extensions suited to the tech stack.
- Cross-platform availability supported work across different operating systems.

Version control was managed using Git for source code management, with GitHub as the remote repository.

- **Key functions:**
  - Tracking changes through commits.
  - Branching and merging to manage new features and bug fixes without affecting the main codebase.
  - Collaboration and code reviews with peers and supervisors.
  - Hosting the project repository remotely to maintain backups and facilitate continuous integration.

- **Advantages:**
    - Prevented loss of code and simplified collaborative work.
    - Allowed for tracking the history of the project's evolution.

AWS provided the infrastructure and cloud services essential for hosting and deploying the application.

Services used:

- **EC2:** For hosting the Django web application server.
- **S3:** Intended for storage of static files and media, including academic documents.
- **CloudWatch:** For monitoring server health and logs.
- **IAM:** Managing secure user permissions and roles.
- **Benefits:**
    - Scalability to accommodate increasing user demand.
    - High availability and fault tolerance.
    - Secure cloud environment adhering to industry standards.

Containerization and Virtualization

Although primarily focused on development and deployment using AWS EC2, containerizatio tools like **Docker** were explored for creating consistent development environments.

- **Purpose:**
    - Ensuring the application runs consistently across development, testing, and production environments.
    - Simplifying deployment by packaging the application and its dependencies.
- **Advantages:**
    - Portability of the application.
    - Isolation of different application services.

    - API behavior.
    - Automating API testing sequences.
- **Role in the project:**
    - Debugging API integrations between frontend and backend.
    - Ensuring secure and reliable data communication.

CHAPTER 4
METHODOLOGY
4.1 APPROACH AND TECHNIQUES USED

4.1 Approach and Techniques Used

During my internship at Jain University, the development of the Knowledge Management Architecture (KMA) system and the Jain Event platform involved a systematic and methodical approach. This chapter outlines the overall methodology adopted, highlighting the development lifecycle, design principles, and techniques employed to achieve project goals efficiently.

The project was executed using an **Agile methodology**, emphasizing iterative progress through continuous feedback and collaboration with supervisors and team members.**:**

Quick adaptation to changing needs and priorities.Early delivery of working modules for feedback.

Benefits realized:

Enhanced flexibility in incorporating new features.Improved communication between stakeholders.Incremental improvements ensured steady progress without major setbacks.Before initiating development, thorough **requirement gathering and analysis** were conductedMeetings with academic staff and IT department to identify core functionalities.Defining roles and permissions for different users – Admin, Faculty, HODs.

Documenting Functional Requirements:

Features such as event creation, academic document management, user authentication, and reporting.

Assessing available technologies and resources for implementation.

Setting milestones and deadlines aligned with the internship duration.

Task allocation and time management schedules.A modular and scalable system design was adopted to support future expansion.

Model-View-Template (MVT) Pattern:

Leveraged Django’s MVT architecture to organize code and separate concerns.

Database Schema Design:

Designed relational schemas for users, events, academic modules, and document storage.

mplemented foreign key relationships to maintain data integrity.

API Design:

RESTful APIs were designed for communication between frontend and backend.

Used JSON format for data exchange ensuring lightweight and readable payloads.

Security Considerations:

Role-based access control to protect sensitive academic data.

Implementation of secure authentication mechanisms.

Prevention of common web vulnerabilities (e.g., CSRF, SQL Injection).

4.1.4 Development Techniques

The development phase was conducted through a combination of manual coding, debugging, and testing, following best practices.

Backend Development:

- Using Python and Django, the server-side logic was coded for handling requests, managing the database, and processing data.
- Extensive use of Django ORM for querying the PostgreSQL database.

Frontend Development:

- Utilized HTML5, CSS3, and Bootstrap to create responsive and user-friendly interfaces.
- JavaScript was employed for enhancing interactivity.

Version Control: Git was used to maintain code versions, enabling safe experimentation and collaboration.**Continuous Testing:**Unit tests and integration tests ensured that new code did not break existing functionality.Postman was used to validate API endpoints.

Debugging: Systematic debugging using logs and error messages helped identify and fix issues efficiently.

4.1.5 Deployment and Cloud Integration

Deploying the application on the cloud was an essential aspect of the project to make it accessible to end-users.

- **AWS EC2 Instances:**Deployed the Django application on Amazon EC2 instances configured with appropriate security groups.
- **Static and Media File Handling:**Managed via AWS S3 buckets for scalability and reliability.
- **Monitoring and Maintenance:**Used AWS CloudWatch to monitor server uptime and performance metrics.Scheduled backups and database maintenance tasks to ensure data safety.

4.1.6 Collaboration and Communication Techniques

Effective communication was maintained throughout the internship to ensure project alignment.

- **Daily Stand-ups and Weekly Reviews:**
  - Shared updates on progress, blockers, and upcoming tasks.
- **Documentation:**
  - Maintained clear and detailed documentation for each module developed.
- **Feedback Incorporation:**
  - Actively sought feedback from supervisors and peers to improve code quality and project features.

techniques were employed:

- **Root Cause Analysis:**
  - Analyzed errors in logs and stack traces to identify underlying issues.
- **Research and Learning:**
  - Consulted official documentation, forums, and tutorials to find solutions.
- **Pair Programming and Code Reviews:**
  - Collaborated with peers to review code and suggest improvements.
- **Task Prioritization:**
  - Used tools like Trello or simple task lists to prioritize critical bugs and features.

The methodology employed combined Agile principles, modular design, and modern development practices to ensure the successful completion of the internship project. This structured approach facilitated smooth progress despite challenges and allowed for the delivery of a functional and scalable web application supporting academic management at Jain University.

4.2 WORK BREAKDOWN

The project undertaken during the internship involved multiple stages and tasks, organized systematically to ensure efficient progress and timely completion. The work breakdown structure (WBS) outlines the division of the entire project into smaller, manageable components. This segmentation helped in clear task allocation, tracking progress, and achieving milestones effectively.

4.2.1 Project Planning and Requirement Gathering

- **Understanding Project Scope:**
  - Initial meetings with supervisors and stakeholders to understand the vision and goals of the KMA and Jain Event platform.
  - Defining user roles such as Admin, Faculty, HOD, and general users.
- **Requirement Documentation:**
  - Gathering functional and non-functional requirements.
  - Listing key features including event management, academic resource handling, user authentication, and permission management.
- **Technology Stack Finalization:**
  - Deciding on Python, Django, PostgreSQL, AWS, and frontend tools based on project needs.

4.2.2 System Design

- **Database Design:**
  - Creating ER diagrams and relational schema for users, events, academic modules, specializations, semesters, and documents.
  - Establishing relationships and constraints to ensure data integrity.
- **Application Architecture:**
  - Designing the backend using Django's MVT architecture.
  - Planning REST API endpoints for frontend-backend communication.
- **User Interface Design:**
  - Sketching wireframes for key pages such as Home, Event Listing, Admin Dashboard, and Document Upload forms.
  - Selecting frontend frameworks and CSS libraries.

4.2.3 Backend Development

- **User Authentication and Authorization:**
  - Implementing login/logout, role-based access control, and permission settings.
- **Core Features Development:**
  - Event management modules allowing admins to add, update, and delete events.
  - Academic resource management including uploading, categorizing, and accessing syllabus, course matrices, and results.
  - Integration of user roles with specific access rights.
- **Database Integration:**
  - Configuring PostgreSQL and connecting it via Django ORM.
  - Writing complex queries for efficient data retrieval and updates.

4.2.4 Frontend Development

- **Static Pages:**
  - Building responsive pages using HTML5, CSS3, and Bootstrap.
  - Creating navigation menus, forms, and tables for data display.
- **Dynamic Content Loading:**
  - Using JavaScript for dynamic updates and user interactions.
  - Implementing event handlers for form submissions and page navigation.
- **User Experience Enhancements:**
  - Adding validation feedback, loading indicators, and error messages.
  - Ensuring cross-browser compatibility and mobile responsiveness.

4.2.5 API Development and Testing

- **REST API Implementation:**
  - Designing and coding endpoints to support CRUD operations.
  - Securing APIs with authentication tokens and permission checks.
- **Testing APIs:**
  - Using Postman to test API responses for various request types.
  - Debugging and fixing issues related to data handling and security.
- **Integration:**
  - Connecting frontend UI components to backend APIs to fetch and display data dynamically.

4.2.6 Cloud Deployment

- **AWS EC2 Setup:**
  - Configuring EC2 instances to host the Django application.
  - Setting up security groups, firewalls, and domain routing.

- **Static and Media File Handling:**
  - Configuring AWS S3 buckets to store and serve static assets and user-uploaded documents.
- **Monitoring and Maintenance:**
  - Setting up AWS CloudWatch for server health monitoring.
  - Establishing backup routines for database and application data.

4.2.7 Documentation and Reporting

- **Code Documentation:**
  - Writing comments and docstrings for functions, classes, and modules.
  - Maintaining a README file with setup instructions and project overview.
- **User Manuals:**
  - Preparing simple guides for admins and users to navigate the system.
- **Progress Reports:**
  - Regular updates to supervisors highlighting completed tasks and upcoming work.
  - Documenting challenges faced and solutions implemented.

4.2.8 Testing and Debugging

- **Unit Testing:**
  - Writing and executing tests for individual components to ensure correctness.
- **Integration Testing:**
  - Validating the interoperability between frontend, backend, and database.
- **Bug Fixing:**
  - Tracking bugs using issue logs.
  - Prioritizing and resolving bugs based on severity.
- **Performance Optimization:**
  - Refining database queries and API responses to reduce latency.
  - Optimizing frontend assets for faster load times.

4.2.9 Final Review and Deployment

- **Feature Finalization:**
  - Completing all pending features and enhancements.
- **System Testing:**
  - Conducting thorough testing in the production-like environment.
- **User Acceptance Testing (UAT):**
  - Collecting feedback from supervisors and sample users.
- **Production Deployment:**
  - Deploying the final version of the system on AWS.
- **Training and Handover:**

    - Conducting user training sessions.
    - Handing over project documentation and credentials.

The structured breakdown of the work ensured that the project was developed in a phased manner, allowing for focused efforts on each aspect of the system. This approach facilitated clear progress tracking and ensured timely delivery while maintaining the quality and robustness of the final product.planning and time-bound milestones were crucial for managing project progress and ensuring successful completion. The project involved both backend and frontend development, integration with AWS for deployment, and system documentation. Each phase was strategically divided into specific milestones, aligned with deadlines, and regularly reviewed to track accomplishments and adjust the roadmap accordingly.

4.3.1 Initial Phase (Week 1 – Week 2)

Milestone 1: Orientation and Requirement Gathering

Activities Completed:

- Understanding departmental objectives and user roles (Admin, HOD, Faculty).
- Meetings with supervisors to finalize scope.
- Analysis of requirements for the KMA system and Jain Event Management module.

- **Outcome:** Finalized a list of core features and drafted the development roadmap.

Milestone 2: Technology Stack Setup

- **Activities Completed:**
    - Installed and configured Django, PostgreSQL, and virtual environments.
    - Set up project structure and repositories using Git.
- **Outcome:** Development environment was successfully established and version control initiated.

4.3.2 Development Phase I (Week 3 – Week 5)

Milestone 3: Backend Database Design and API Development

Activities Completed:

- Designed relational schema for events, users, departments, documents, and permissions.
- Developed APIs using Django REST Framework.
- Implemented role-based access control for Admins, Faculty, and HODs.

- **Outcome:** Core backend models and API endpoints were created and tested using Postman.

Milestone 4: Initial Frontend Development and Integration

- **Activities Completed:**
    - Built static and dynamic templates using HTML, CSS, Bootstrap, and basic JavaScript.
    - Integrated backend APIs with frontend forms and pages.
    - Developed Admin Dashboard for event and document management.
- **Outcome:** Working prototype with backend–frontend integration for key features.

4.3.3 Development Phase II (Week 6 – Week 8)

Milestone 5: Cloud Deployment and Media Management

Activities Completed:

- Set up AWS EC2 instances and deployed Django project.
- Configured AWS S3 for static/media file handling.
- Secured the application using HTTPS and IAM roles.

- **Outcome:** Deployed version of the application accessible via AWS.

Milestone 6: User Authentication and Permissions

- **Start Date:** March 19, 2025
- **End Date:** March 25, 2025
- **Activities Completed:**
    - Implemented Django’s authentication system.
    - Developed permission modules allowing different user levels to view/edit features.
- **Outcome:** Enhanced system security and access control.

CHAPTER 5
WORK/PROJECT DETAILS

5.1 TASK BREAKDOWN

The internship project titled **"Knowledge Management Architecture (KMA) & Jain Event Management System"** was developed to centralize and simplify the management of academic events, resources, and user-specific information within Jain University. To ensure successful completion of this full-stack web development project, the work was divided into specific tasks that followed a structured and logical progression. These tasks covered all phases from conceptualization to deployment.

Phase 1: Requirement Analysis and Planning

Objective: Understand the expectations and determine system requirements
Tasks:

- Attended onboarding sessions with the Office of Academic Affairs.
- Understood the core functional areas: academic calendar, event hosting, document uploads, and role-based dashboards.
- Created the Software Requirements Specification (SRS) document.
- Prepared project timelines and deliverables aligned with internship duration.

Phase 2: Database Design and Model Structuring

Objective: Create a robust backend using Django models and PostgreSQL
Tasks:

- Designed relational models for Users, Roles (Admin, HOD, Faculty), Departments, Events, Documents, and Specializations.
- Used Django ORM to define models and migrate schemas.
- Implemented foreign key relationships and indexing for performance optimization.
- Validated the schema with sample data entries and test queries.

Phase 3: Backend Development using Django

Objective: Implement backend logic and REST APIs for core operations
Tasks:

- Built authentication and user login/logout system using Django’s auth module.
- Developed RESTful APIs to manage:
    - Event CRUD operations.

    - File uploads (syllabus, result, course matrix).
    - Specializations, modules, semesters, and academic roles.
- Implemented role-based access control to restrict functionalities per user type.
- Used Django Admin for quick content management by superusers.

Phase 4: Frontend Development

Objective: Design interactive, responsive UI for all user types
Tasks:

- Designed pages for:
    - Home
    - Admin dashboard
    - Event listing and addition
    - Academic document uploads
- Used HTML5, CSS3, Bootstrap, and JavaScript for responsive layout.
- Created dynamic forms and tabs for year-wise content filtering.
- Integrated frontend forms with backend APIs using AJAX and fetch API.

Phase 5: Testing and Debugging

Objective: Ensure application stability and security
Tasks:

- Wrote unit tests for APIs and critical functions.
- Manually tested all UI components and validated form inputs.
- Used Postman to test backend endpoints under different scenarios.
- Resolved issues related to:
    - File validation (PDF format checks)
    - Duplicate entries and empty fields
    - Unauthorized access and session timeout
- Optimized slow SQL queries and redundant data loading.

1. CHALLENGES ENCOUNTERED

During the internship tenure at the Office of Academic Affairs, Jain University, and while working on the project titled "Knowledge Management Architecture (KMA) and Jain Event Management System," **various challenges—both technical and non-technical—were encountered. These challenges contributed significantly to my learning curve and provided practical insights into the real-world software development process.**

I. Technical Challenges

- Designing a scalable and normalized relational database structure for managing departments, events, user roles, academic documents, and multiple file uploads was a challenging task.
- The need to ensure data integrity while also providing flexibility for admin operations added complexity.
- **Solution Implemented:** Followed entity-relationship design principles and used Django’s ORM to simplify relationship handling between models.Integrating frontend with backend APIs using JavaScript (AJAX/fetch) posed multiple bugs—such as incorrect payloads, 403/404 errors, and inconsistent data rendering.

- Some forms failed to handle file uploads correctly during testing phases.
- **Solution Implemented:** Used developer tools and Postman to debug and refine API request-response cycles. File handling was optimized using Django’s request.FILES and FileField..Implementing different levels of user roles (Admin, HOD, Faculty) with precise permissions for data visibility and editing was a complex requirement.

- Mistakes in permission logic initially allowed unauthorized users to access restricted content.
- **Solution Implemented:** Used Django's built-in authentication and custom decorators to restrict views and actions based on user roles.During the deployment process, issues such as database connection errors, improper static/media file rendering, and misconfigured settings.py files caused major setbacks.

- Integrating AWS S3 and EC2 with the Django project and configuring HTTPS also required extensive trial and error.
- **Solution Implemented:** Followed AWS documentation, community forums, and best practices. Added logging to trace issues in production.

5.2 SOLUTIONS IMPLEMENTED

Throughout the course of the internship, multiple challenges were encountered across technical, managerial, and operational areas. Each challenge served as an opportunity for problem-solving and learning. This section outlines the **strategic, technical, and procedural solutions** that were implemented to ensure project progress, stability, and success..

Added support for multiple file uploads using formset_factory() and JavaScript-based dynamic form fields.

CHAPTER 6
LEARNING OUTCOMES

6.1 TECHNICAL SKILLS GAINED

During my 3-month internship as a Web Developer Intern at Jain University's Office of Academic Affairs, I had the opportunity to work extensively on the **Knowledge Management Architecture (KMA)** and **Jain Event Management System** projects. This experience significantly enhanced my technical competencies across several tools, languages, and frameworks.Gained proficiency in writing clean, modular, and scalable Python code for backend development.

- Applied object-oriented programming concepts to build maintainable Django models logic.
- Wrote custom utility scripts for data migration, file handling, and automated tasks.Developed web applications using Django, focusing on both **function-based** and **class-based views**.

- Worked with **Django ORM** to manage relational databases effectively without raw SQL.
- Customized the Django **Admin Panel** to handle complex form submissions, user authentication, and role-based permissions.
- Managed **media files**, static content, and integrated user-uploaded documents within the portalDesigned responsive and user-friendly interfaces using **HTML5**, **CSS3**, and **Bootstrap 5**.

- Applied **JavaScript** and **AJAX** to fetch data asynchronously, improving user experience.
- Implemented dynamic frontend behavior such as modal pop-ups, dropdowns, and tabbed navigation for ease of access to academic resources.Built RESTful APIs using **Django REST Framework** for frontend-backend communication.

- Integrated APIs to enable file uploads, form submissions, event creation, and real-time data display.
- Used tools like **Postman** to test and debug API endpoints before deployment.

5. Database Management

- Designed and managed **relational databases** using **PostgreSQL** with Django.
- Created relationships among models such as Users, Departments, Specializations, Semesters, and Modules.
- Gained experience in data migration, indexing, and query optimization for performance improvement.Deployed the Django application on an **AWS EC2 instance**, managing server configuration and environment setup.

- Hosted static and media files using **AWS S3**, enabling scalable storage solutions.
- Managed security using **SSH access**, security groups, and IAM roles.
- Configured **Gunicorn** and **Nginx** for a production-ready web environment.Used Git for version control, including **branching**, **merging**, and **resolving conflicts**.

- Maintained a structured repository with regular commits, improving collaboration and traceability.
- Followed best practices in commit messaging and repository structure to ensure clarity and Learned systematic debugging using Django's error logs, Python's traceback, and browser developer tools.

- Wrote unit tests for critical backend functions.
- Conducted manual and exploratory testing to ensure app stability and proper role-based data visibilityEnabled **CSRF protection**, **password hashing**, and **secure form handling** to ensure the application was resistant to common vulnerabilities.

- Restricted access to sensitive modules using Django's permission system and middleware checks.Wrote technical documentation for database schemas, API usage, and deployment procedures.

- Automated deployment steps using shell scripts for consistency across environments.By the end of this internship, I had not only gained hands-on experience in full-stack development using Python and Django, but also developed strong deployment management skills. This internship bridged the gap between theoretical learning world application, preparing me for more complex, production-grade software development in the future.

6.2 SOFT SKILLS IMPROVED

In addition to technical growth, my internship at Jain University's Office of Academic Affairs significantly contributed to the development of various **soft skills** that are essential in a professional work environment. These soft skills not only helped me function efficiently withi a team but also enabled me to approach tasks with confidence, responsibility, and clarity.

Professional Communication: I learned how to interact with supervisors, academic staff, and team members

in a clear, respectful, and professional manner. This included writing formal emails, giving updates during meetings, and documenting project progress.

Technical to Non-Technical Translation: One of the most valuable skills I developed was th ability to explain technical concepts (such as APIs, databases, and deployment procedures) to non-technical stakeholders in a way they could understand.

- **Feedback Handling:** I became more comfortable receiving constructive criticism and using it to improve my work, especially in design reviews and feature implementation meetings.**Cross-Department Coordination:** I regularly collaborated with different departments like Event Management, Academic Records, and IT Support. how to align technical work with institutional goals.

- **Responsibility Sharing:** I worked in sync with other interns and senior developers, ofte contributing ideas in brainstorming sessions and pairing up to solve complex tasks.
- **Team Productivity Tools:** I improved my usage of tools like Trello, Google Docs, and Slack for effective team communication and task tracking.**Task Prioritization:** With multiple deadlines and feature requests, I learned how to prioritize tasks based on urgency and impact, using structured to-do lists and Kanban boards.

- **Deadline Discipline:** I learned how to break down large goals into smaller deliverables and commit to deadlines while maintaining quality.
- **Meeting Deadlines Under Pressure:** Managing the frontend UI while simultaneously handling Django views and testing helped me work under time constraints and still deliver consistent results.

4. Problem-Solving Attitude

- **Analytical Thinking:** I trained myself to identify the root causes of bugs and integratio issues, analyze logs, and find step-by-step solutions.
- **Independence:** Although guidance was available, I was often encouraged to first try solving issues independently. This boosted my confidence and encouraged a problem-solving mindset.

- **Creative Solutions:** I learned to think outside the box when certain features couldn't be implemented as expected due to technical constraints or time limits.**Quick Learning:** I had to quickly learn tools like AWS, Git, and Django REST Framework—most of which were not part of my prior curriculum. This improved my ability to adapt to new technologies and environments.

- **Dealing with Change:** Requirement changes and sudden priority shifts were common. I became more flexible in adjusting my plans while keeping the overall project objective intact.

- **Proactive Contribution:** I proposed new UI enhancements, recommended better database structures, and took the initiative to lead some documentation efforts.
- **Ownership:** I took responsibility for bugs, delays, and technical missteps, and worked hard to correct them promptly—building a sense of ownership for the project.
- **Dependability:** My mentor and teammates began to rely on me for tasks involving authentication flows, document uploads, and role management.

6.3 INSIGHTS GAINED FROM THE INTERNSHIP

My internship experience at **Jain University – Office of Academic Affairs** has been truly transformative. Beyond acquiring technical knowledge and refining soft skills, this opportunity provided me with deep insights into how real-world software projects are conceived, managed, and delivered in an institutional environment. These insights extend beyond textbook learning and have reshaped my understanding of professional responsibilities, project lifecycles, teamwork, and career aspirations.

One of the most valuable takeaways was gaining firsthand experience in how full-scale web applications are built from the ground up. The **Knowledge Management Architecture**

(KMA) and **Jain Event System** weren't just theoretical projects; they involved live databases, real users, and dynamic institutional needs. This helped me understand:

- The importance of planning before coding.
- The value of maintaining clean, modular, and scalable code.
- The need for aligning development with user requirements and feedbackAlthough I had prior academic exposure to Python, databases, and web technologies, working on live applications helped me connect theoretical concepts with practical implementation. Key examples include:

- Using Django's ORM instead of raw SQL to handle data manipulation securely.
- Understanding how deployment environments (like AWS) differ from local development setups.

- Realizing the importance of security, documentation, and performance, which are often overlooked in academic projects.Effective communication turned out to be just as important as coding. Whether it was discussing features with the team or explaining a bug during meetings, I learned how crucial it is to clearly express ideas, ask questions, and maintain documentation. Key insights included:

- Daily stand-up meetings keep the team aligned and productive.
- Peer feedback significantly improves code quality and user experience.
- Collaborative tools (GitHub, Trello, Google Docs) play a vital role in managing tasks Internship tasks were often fast-paced and had shifting priorities. I learned to manage time effectively through:

I also learned that flexibility is a must-have skill in software development, especially when integrating with external APIs or deploying across cloud services.Working on an academic portal gave me a new perspective on designing systems that are intuitive and accessible for diverse user groups including admin staff, faculty, and students. I now understand:

The deployment phase of the project was especially eye-opening. From configuring AWS EC2 and S3 to understanding how NGINX serves content, I gained exposure to deployment

t practices used in real-world projects. This broadened my horizon beyond coding and into the realm of infrastructure and DevOps.lies. I discovered that I enjoy backend development, deployment, and systems integration. It also motivated me to pursue further learning in cloud architecture and scalable web development.

CHAPTER 7

RESULTS AND CONTRIBUTIONS

7.1 DELIVERABLES

During the course of my internship at **Jain University's Office of Academic Affairs**, I was entrusted with tasks that contributed to the development, improvement, and deployment of two major web-based platforms: the **Knowledge Management Architecture (KMA)** system and the **Jain Event Portal**. These platforms are designed to enhance academic data accessibility and event management within the university.

Below is a detailed summary of the key deliverables that I successfully completed over the internship duration:

KMA (Knowledge Management Architecture) System

Designed models for academic departments, faculty, specializations, semesters, modules, and course files.Created Django views and serializers to handle user authentication, document management, and academic records retrieval.Integrated Django admin panel for department-level access and secure data entry.Developed functionality for role-based access (Admin, Director, HOD, Faculty).Implemented permission-based operations for data upload download, and deletion.Created file upload forms for managing syllabus, course matrices, TLEP, and result files.Configured file handling logic for efficient data storage and secure access.Enabled dynamic filtering of uploaded documents based on year, semester, and specialization.Integrated Django's in-built authentication system.Added login portals for Admin and Faculty.Implemented user permission controls using Django's permissions and groups module. **Jain Event Portal**

- Developed backend logic for creating, updating, and deleting academic and extracurricular events.
- Integrated event categories (e.g., workshops, guest lectures, seminars, and cultural programs).
- Designed admin panels for adding events, managing categories, and assigning users to roles.
- Created forms for uploading banners, event brochures, and participant data.
- Built responsive user interfaces using HTML, CSS, and basic JavaScript.

- Designed intuitive home and event listing pages accessible to students and staff.

Cloud Deployment and Hosting (AWS)

- Deployed the Django-based KMA system on an AWS EC2 instance.
- Configured environment variables, dependencies, and WSGI application using Gunicorn and NGINX.

- Configured AWS S3 for hosting uploaded files such as syllabus PDFs and event images.
- Ensured media access and security using IAM roles and proper bucket policies.

- Wrote startup scripts for restarting services on instance reboot.
- Configured automated backup options for the PostgreSQL database (optional for future implementation).

ACHIEVEMENTS AND IMPACTS

During my three-month internship as a **Web Developer Intern** in the **Office of Academic Affairs at Jain University**, I had the opportunity to work on critical systems that enhanced institutional productivity, data accessibility, and user experience. The successful completion of project milestones led to several key **achievements** and **tangible impacts**, both for the institution and my personal and professional growth.1. Successful Implementation of the Knowledge Management Architecture (KMA).Developed a centralized academic platform that enabled secure upload, storage, and retrieval of syllabus, course matrices, results, and TLEP documents.The platform served as a **one-click access portal** for faculty, HODs, and Directors, improving the speed and accuracy of academic data handling.Created an intuitive system to **manage university events** by category and department.Enabled real-time event creation, updates, and display across the institution, reducing dependency on manual notices and emails.Implemented a fine-grained permission system that ensures only authorized personnel can perform certain actions.Prevented unauthorized access and data breaches, enhancing the **security** of the academic records system.Successfully deployed the KMA system on **AWS EC2**, simulating real-world deployment practices.Ensured system uptime, fast loading times, and data availability for remote users.Coordinated with technical and non-technical departments, collecting requirements and training staff on system usage.

B. POSITIVE IMPACTS

On Jain University

- **Efficiency**: Drastically reduced time taken for uploading and accessing academic documents.

- **Transparency**: Created an audit trail of who uploaded or modified what, aiding in accountability.
- **Digital Transformation**: Contributed to the University's vision of transitioning from manual to **automated, paperless operations**.
- **Faculty Empowerment**: Faculty and HODs were able to manage their documents and events independently without IT team dependency.

On Myself

- **Skill Development**: Enhanced skills in Python, Django, AWS, Git, React basics, and database management.
- **Professional Confidence**: Gained confidence to handle projects independently and take ownership of tasks.
- **Career Direction**: Strengthened my interest in full-stack web development and cloud infrastructure.
- **Problem Solving**: Learned to approach technical bugs and performance issues systematically and with patience.
- **Communication**: Improved technical writing and verbal communication by
- documenting systems and interacting with stakeholders.

RECOGNITION & FEEDBACK

- **Supervisor Appreciation**: Recognized for efficient code structure, disciplined task management, and proactive attitude.
- **Team Feedback**: Colleagues appreciated my **quick debugging skills** and adaptability in understanding new technologies.
- **Impact Assessment**: My contributions are now used by multiple departments and may form the foundation for future enhancements in the Academic Portal.

.

7.2 FEEDBACK FROM THE ORGANIZATION

Throughout the course of my internship at **Jain University**, I consistently engaged with supervisors, team members, and departmental staff to ensure that my contributions aligned with the goals of the **Office of Academic Affairs**. The feedback I received from the organization was both encouraging and constructive, highlighting my strengths and offering guidance for future development.

My immediate supervisor and project manager recognized the value of my contributions to the **Knowledge Management Architecture (KMA)** system and the **Jain Event Portal**. The following points summarize their feedback:

- **Timely Delivery**: Commended for completing assigned modules and deployment tasks ahead of deadlines.
- **Code Quality**: Praised for writing clean, modular, and well-documented code that adhered to industry best practices.
- **Proactiveness**: Acknowledged for taking the initiative to suggest feature improvements and anticipate potential system requirements.
- **Problem Solving**: Appreciated for resolving bugs quickly during critical development phases, particularly in database integration and access control.
- **Documentation**: Advised to further improve documentation for future maintainers and users, especially for non-technical stakeholders.I regularly collaborated with fellow interns, technical staff, and administrative users. The peer feedback was informal yet highly insightful:

- **Team Collaboration**: Recognized for maintaining clear communication and for being a reliable team player during group development tasks.
- **Adaptability**: Appreciated for adapting to changing project requirements and learning new technologies (like AWS deployment) on the go.
- **UI Suggestions**: Encouraged to explore more advanced frontend tools like React and improve UI/UX for better user engagement.
- **Training Sessions**: Received thanks for offering quick walkthroughs and training assistance to HODs and faculty unfamiliar with the admin system.

CHAPTER 8
CONCLUSION

8.1 SUMMARY OF INTERNSHIP EXPERIENCE

The internship has been a transformative journey that significantly contributed to both my technical proficiency and professional growth. Starting with a foundational understanding of web development, cloud deployment, and network operations, I was able to apply theoretical knowledge in real-world scenarios, gaining invaluable hands-on experience. This practical exposure enabled me to develop scalable web applications using Python and Django, deploy these applications on AWS, and manage network access and monitoring using industry-

standard tools.Throughout the internship, I encountered multiple challenges that tested my problem-solving abilities and adaptability. From debugging complex database migrations to optimizing API performance, I learned to approach problems methodically and leverage available tools effectively. The collaborative environment fostered continuous learning and improved my communication and teamwork skills, essential for any professional setting.My contributions to the Knowledge Management Architecture (KMA) project and the Jain University Event Management portal helped streamline data accessibility and operational efficiency within the academic department. Implementing features for event scheduling, user authentication, and access control not only enhanced the usability of the systems but also deepened my understanding of full-stack development and cloud infrastructure.

Additionally, managing network operations and access permissions strengthened my grasp of security protocols and compliance frameworks, which are critical in today's digital landscape. The experience with tools like Jira, New Relic, and Grafana expanded my technical toolkit, enabling me to monitor system health and performance with confidence.This internship reinforced the importance of continuous learning and professional discipline. I developed a keen sense of accountability, punctuality, and quality assurance in delivering tasks. Regular feedback from mentors helped me refine my approach to coding, documentation, and project management. Moreover, balancing technical work with soft skills such as communication, collaboration, and time management prepared me for future challenges in any dynamic work environment.Looking ahead, I am motivated to build upon this solid foundation by advancing my skills in frontend development, cloud computing, network security, and DevOps. The insights gained during this internship have shaped clear career goals centered on becoming a versatile and impactful technology professional. I plan to pursue relevant certifications and possibly further academic qualifications to stay updated with emerging technologies.

.

8.1 FUTURE CAREER PROSPECTS

My internship experience has significantly shaped my perspective on my future career and has opened up various promising pathways in the field of technology. With hands-on exposure to full-stack web development, cloud computing, network operations, and access management, I have developed a solid foundation that aligns with current industry demands and future trends. This foundation will guide my career trajectory and goals in the coming years.

Exploring Full-Stack Web Development and Cloud Engineering

During my internship, I gained practical skills in Python and Django for backend development, as well as an understanding of deploying applications on cloud platforms like AWS. These competencies make a strong case for pursuing a career in full-stack web development or cloud engineering. As businesses continue to migrate their services to the cloud and emphasize scalable, efficient web applications, professionals skilled in cloud infrastructure and web frameworks are in high demand.

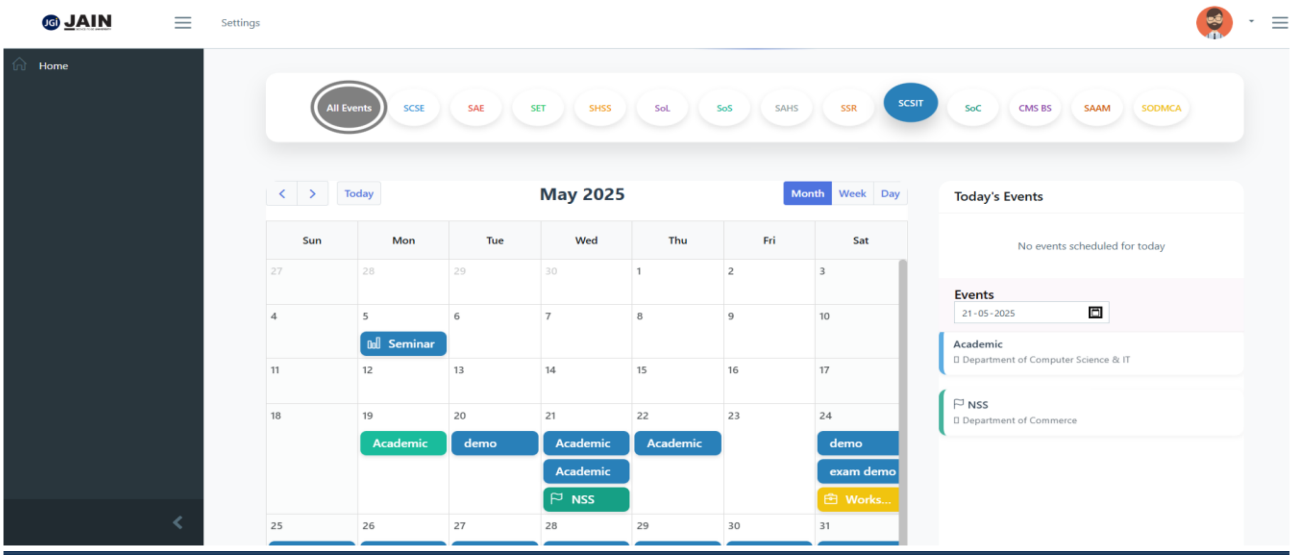
In the near future, I plan to deepen my expertise in modern frontend technologies such as React.js and Angular to complement my backend skills. Mastering these technologies will enable me to become a proficient full-stack developer, capable of designing and deploying end-to-end solutions. Furthermore, pursuing certifications such as AWS Certified Solutions Architect or AWS Developer will enhance my credibility and open doors to roles focused on cloud infrastructure management and DevOps.

Interest in Network Operations and Cybersecurity

My role as a Network Operations Engineer gave me invaluable insight into real-time network monitoring, access management, and issue tracking. This exposure has sparked a keen interest in network administration and cybersecurity. The critical nature of access control and compliance with security frameworks like GDPR has made me appreciate the complexity and importance of safeguarding IT environments.

APPENDICES

Screenshots & Learning Components



May 2025

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5 Seminar	6	7	8	9	10
11	12	13	14	15	16	17
18	19 Academic	20 demo	21 Academic Academic NSS	22 Academic	23	24 demo exam demo Works...
25	26	27	28	29	30	31

Today's Events

No events scheduled for today

Events

21-05-2025

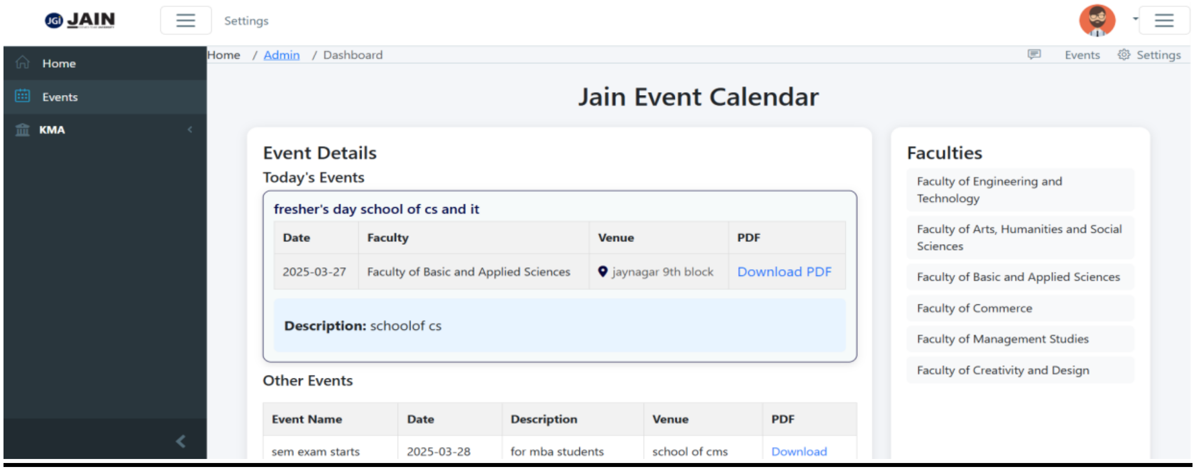
Academic

Department of Computer Science & IT

NSS

Department of Commerce

Figure 1.1 Home Page



Jain Event Calendar

Event Details

Today's Events

fresher's day school of cs and it

Date	Faculty	Venue	PDF
2025-03-27	Faculty of Basic and Applied Sciences	jaynagar 9th block	Download PDF

Description: schoolof cs

Other Events

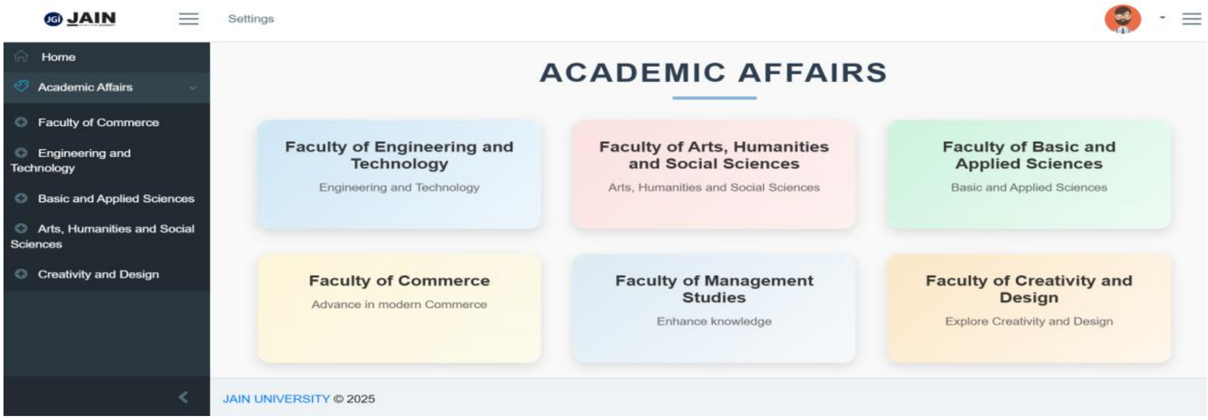
Event Name	Date	Description	Venue	PDF
sem exam starts	2025-03-28	for mba students	school of cms	Download

Faculties

- Faculty of Engineering and Technology
- Faculty of Arts, Humanities and Social Sciences
- Faculty of Basic and Applied Sciences
- Faculty of Commerce
- Faculty of Management Studies
- Faculty of Creativity and Design

Figure 1.2 Event Page

KMA



ACADEMIC AFFAIRS

Faculty of Engineering and Technology
Engineering and Technology

Faculty of Arts, Humanities and Social Sciences
Arts, Humanities and Social Sciences

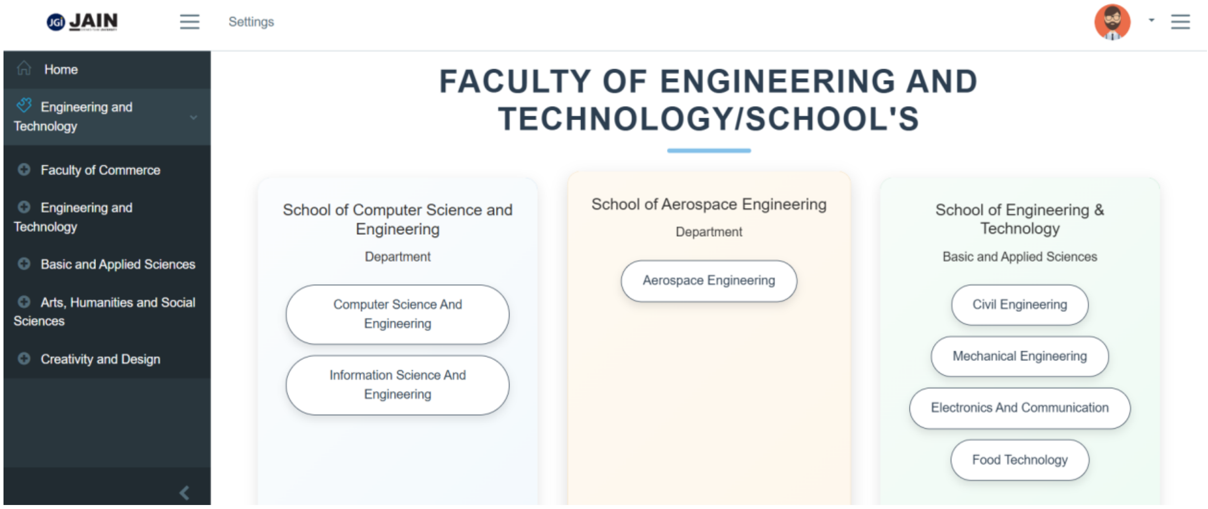
Faculty of Basic and Applied Sciences
Basic and Applied Sciences

Faculty of Commerce
Advance in modern Commerce

Faculty of Management Studies
Enhance knowledge

Faculty of Creativity and Design
Explore Creativity and Design

Figure 1.3 **Academic Affairs-Portal**

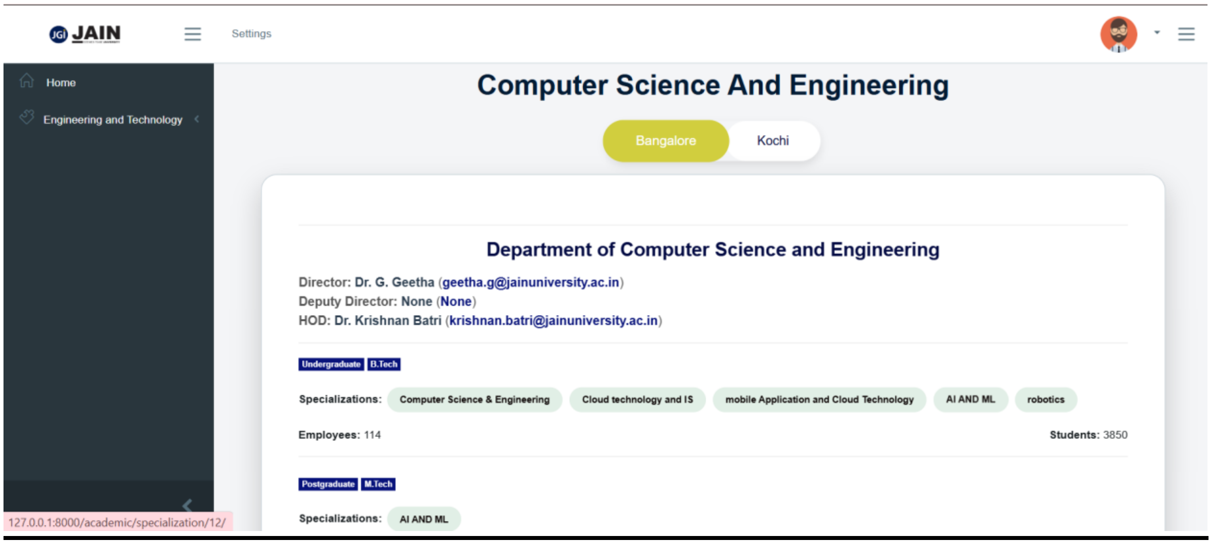

JGI JAIN
Settings
Home
Engineering and Technology
Faculty of Commerce
Engineering and Technology
Basic and Applied Sciences
Arts, Humanities and Social Sciences
Creativity and Design

FACULTY OF ENGINEERING AND TECHNOLOGY/SCHOOL'S

School of Computer Science and Engineering
Department
Computer Science And Engineering
Information Science And Engineering

School of Aerospace Engineering
Department
Aerospace Engineering

School of Engineering & Technology
Basic and Applied Sciences
Civil Engineering
Mechanical Engineering
Electronics And Communication
Food Technology


JAIN
Settings
Home
Engineering and Technology
Bangalore
Kochi
127.0.0.1:8000/academic/specialization/12/

Computer Science And Engineering

Department of Computer Science and Engineering

Director: Dr. G. Geetha (geetha.g@jainuniversity.ac.in)
Deputy Director: None (None)
HOD: Dr. Krishnan Batri (krishnan.batri@jainuniversity.ac.in)

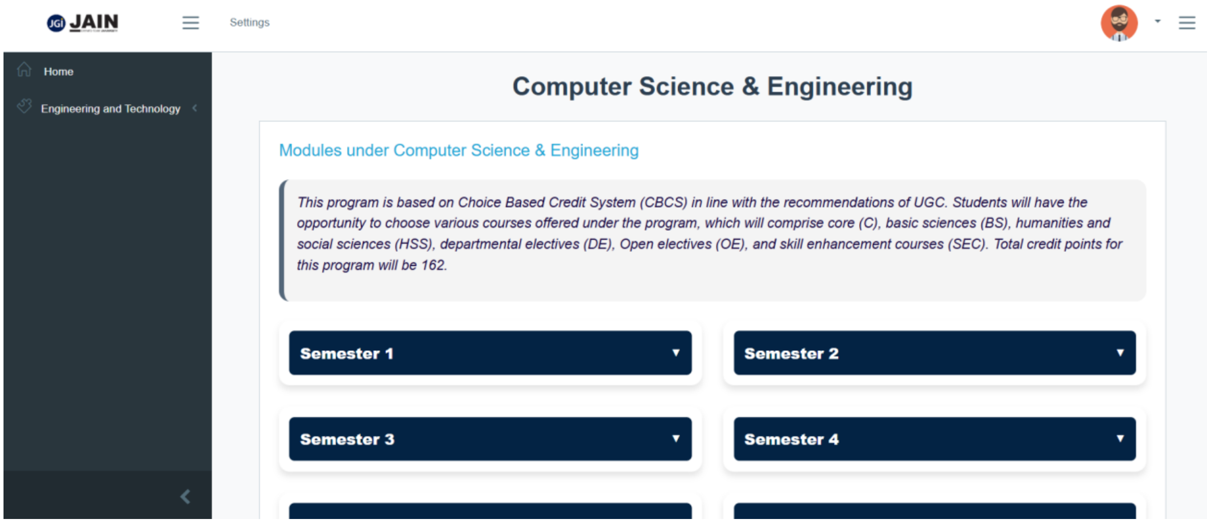
Undergraduate B.Tech

Specializations: Computer Science & Engineering | Cloud technology and IS | mobile Application and Cloud Technology | AI AND ML | robotics

Employees: 114 **Students:** 3850

Postgraduate M.Tech

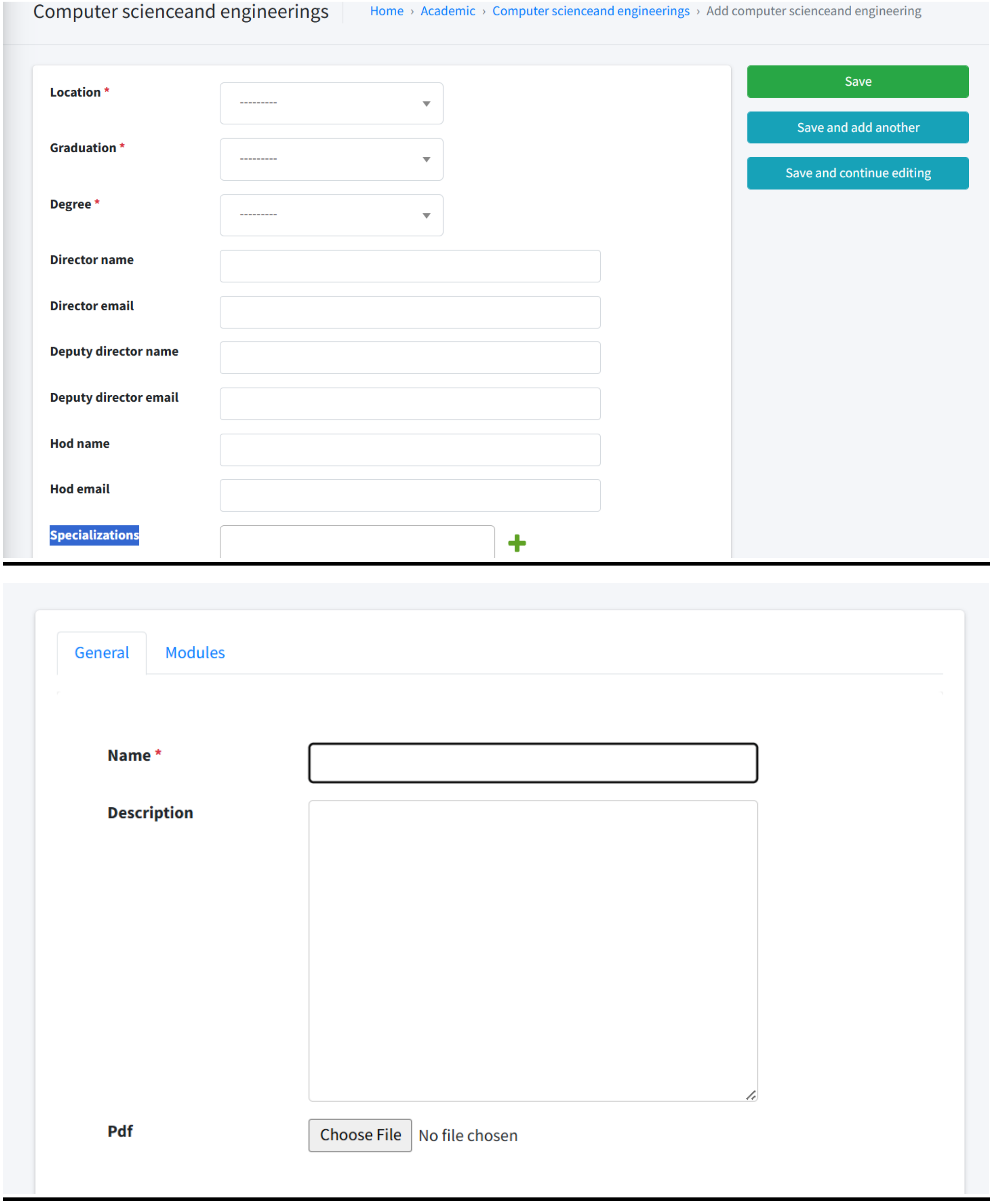
Specializations: AI AND ML



Computer Science & Engineering

Modules under Computer Science & Engineering

This program is based on Choice Based Credit System (CBCS) in line with the recommendations of UGC. Students will have the opportunity to choose various courses offered under the program, which will comprise core (C), basic sciences (BS), humanities and social sciences (HSS), departmental electives (DE), Open electives (OE), and skill enhancement courses (SEC). Total credit points for this program will be 162.



Computer scienceand engineerings

Home › Academic › Computer scienceand engineerings › Add computer scienceand engineering

Location *

Graduation *

Degree *

Director name

Director email

Deputy director name

Deputy director email

Hod name

Hod email

Specializations

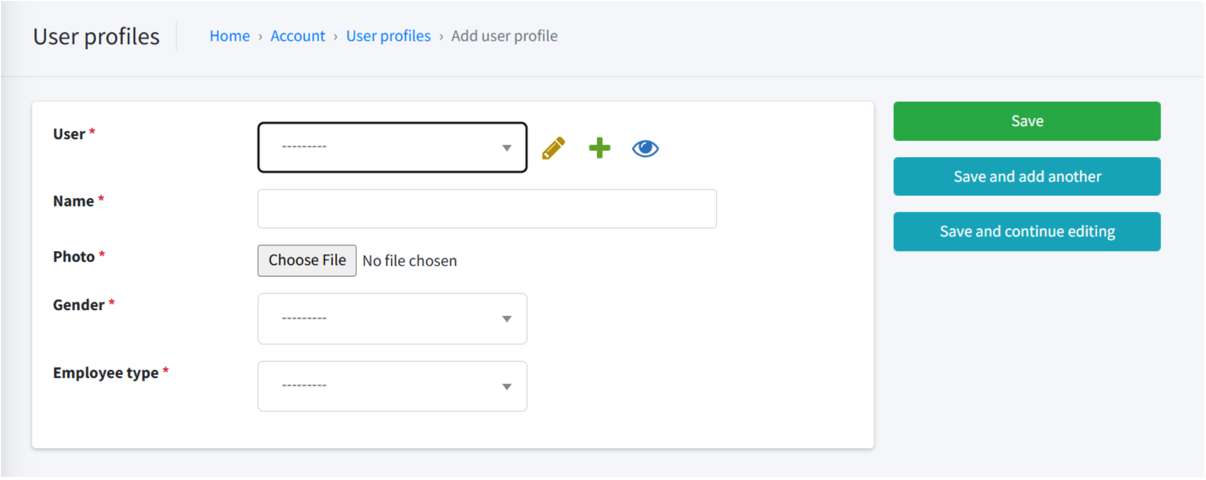
Name *

Description

Pdf

No file chosen

Figure 1.4 Admin -Adding Director, Hod’s, Specializations, Semester, Modules, TLEP



User profiles

Home › Account › User profiles › Add user profile

User * ---------

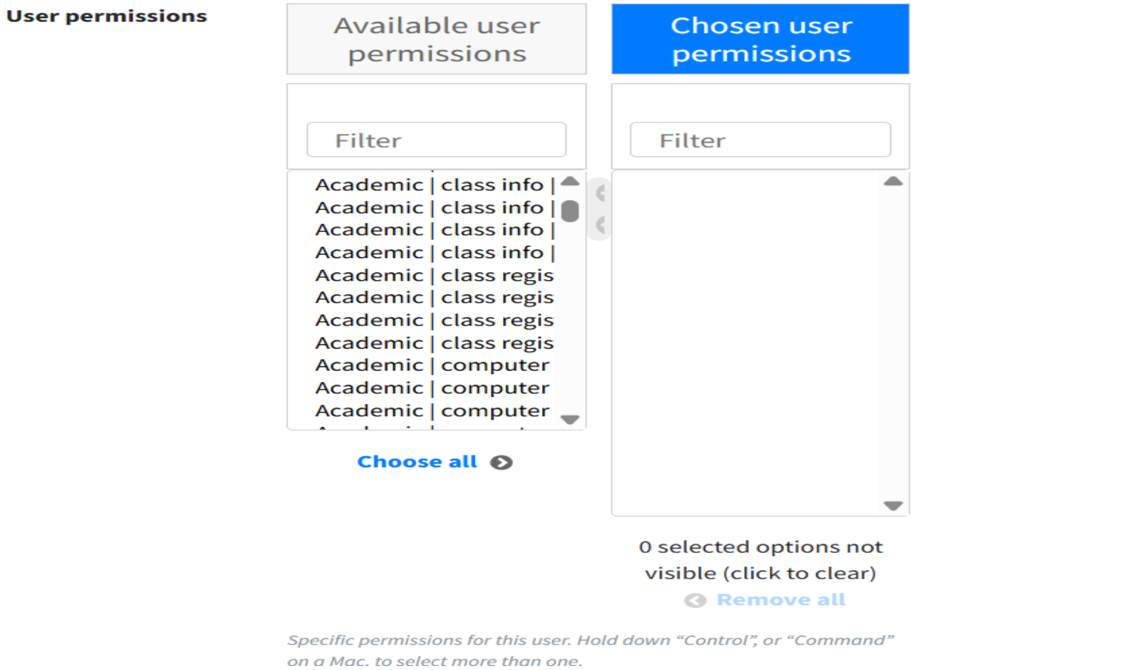
Name *

Photo * Choose File No file chosen

Gender * ---------

Employee type * ---------

Figure 1.5 Admin-User Authentication



User permissions

Available user permissions

Filter

Academic | class info |
Academic | class info |
Academic | class info |
Academic | class info |
Academic | class regis
Academic | class regis
Academic | class regis
Academic | class regis
Academic | computer
Academic | computer
Academic | computer

Choose all

Chosen user permissions

Filter

0 selected options not visible (click to clear)

Remove all

Specific permissions for this user. Hold down "Control", or "Command" on a Mac. to select more than one.

Figure 1.6 Admin-User Permission

REFERENCES

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
2. Chollet, F. (2021). *Deep Learning with Python* (2nd ed.). Manning Publications.
3. OpenCV. (2024). *Image Processing Documentation*. Retrieved from https://docs.opencv.org/
4. Jain University (2024) *Project Handbook* (Internal Document).
5. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
6. Kaggle. (2024). *For dummy data* from https://www.kaggle.com
7. Zhang, Y., et al. (2020). Enhancing Low-Light Images Using Deep Learning. *IEEE Transactions on Image Processing*, 29, 6079-60