**CHAPTER 1**

**INTRODUCTION**

The Faculty Contribution Management System is a comprehensive software solution designed to streamline and enhance the management of faculty contributions within an educational institution. This system aims to provide a centralized platform that facilitates the efficient tracking, evaluation, and documentation of the diverse contributions made by faculty members in areas such as teaching, research, and service.

**1.1** **OVERVIEW**

## A Faculty Contribution Management System is a software application designed to streamline and manage the various contributions and activities of faculty members within an educational institution. The system aims to provide an efficient and transparent way to track, evaluate, and recognize the academic and non-academic contributions made by faculty members. Here is an overview of the key features and components typically found in a Faculty Contribution Management System.

## **1.2 OBJECTIVES**

The objectives of a Faculty Contribution Management System project may vary based on the specific needs and goals of the educational institution implementing it. However, here are some common objectives that such a system might aim to achieve:

**1.2.1 Efficient Tracking of Faculty Contributions:** Enable systematic tracking of various contributions made by faculty members, including teaching, research, publications, conferences, community service, and other relevant activities.

**1.2.2 Performance Evaluation**: Facilitate fair and comprehensive evaluation of faculty performance by providing a transparent and centralized platform for assessing their contributions in different areas.

**1.2.3 Data Accuracy and Transparency:** Ensure accuracy and transparency in recording and reporting faculty contributions to avoid discrepancies and misunderstandings. This could involve automated data entry, validation, and verification mechanisms.

**1.2.4 Resource Allocation**: Assist in effective resource allocation by providing insights into the strengths and areas of improvement for each faculty member, helping administrators make informed decisions.

**1.2.5 Promotion and Tenure Evaluation:** Support the promotion and tenure process by providing a detailed record of a faculty member's contributions over time, aiding in the decision-making process.



**Fig. 1.1 Home Page**

## **1.3 KEY FEATURES**

A Faculty Contribution Management System (FCMS) is a software project designed to streamline and manage the contributions and activities of faculty members in an educational institution. The key features of such a system may include:

**1.3.1 Profile Management:**

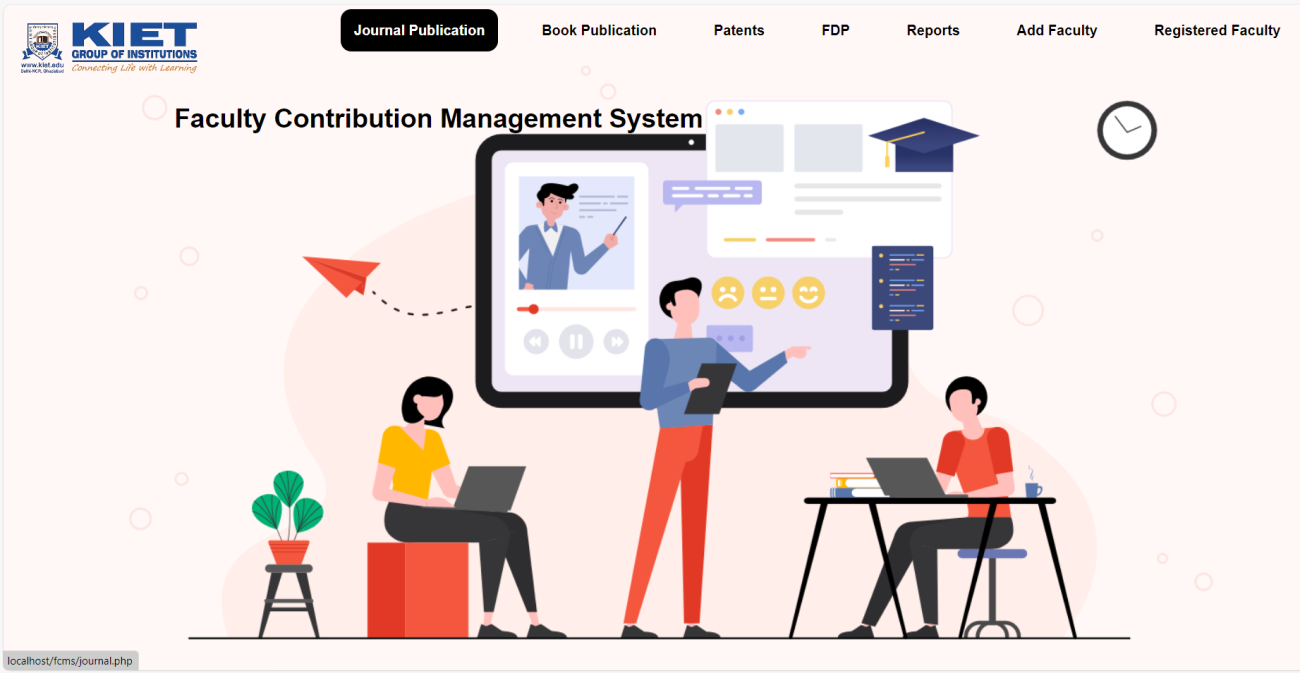
* Faculty members should be able to create and manage their profiles.
* Update personal and professional information, including qualifications, research interests, and contact details.

**1.3.2 Document Management:**

* Upload and manage documents related to research papers, presentations, and other academic contributions.
* Support for various file types and version control.

**1.3.3 Research Tracking:**

* Monitor and record faculty research activities, including publications, conferences, workshops, and collaborative projects.
* Integration with academic databases and citation management systems.



**Fig. 1.2 Select Role**

## **1.4 HARDWARE / SOFTWARE USED IN PROJECT**

The Faculty Contribution Management System will involve a combination of hardware and software components to ensure its development, deployment and functionality. Here is detailed List:

**1.4.1 Server-side Hardware:**

**1. RAM (Random Access Memory):**

* 8GB to 16GB (for moderate-sized application and user load).
* Consider higher capacities (e.g., 32GB or more) for scalability and handling a large number of concurrent users.

**2. ROM (Storage):**

* SSD storage for faster read and write operations.
* Allocate storage based on the application codebase, database size, and media storage requirements.

**3. Processor:**

* Multi-core processor (quad-core or higher) for efficient handling of concurrent user requests.

**4. Operating System:**

* Windows based operating system.

**5. Network Equipment:**

* Network infrastructure to facilitate secure data transfer between users and the server.

**1.4.2 Database Server:**

**1. RAM:**

* 16GB or more for efficient handling of concurrent database queries.

**2. ROM(Storage):**

* SSD storage for faster data retrieval.
* Allocate storage based on the anticipated size of the database and data storage needs.

**3. Processor:**

* Multi-core processor with sufficient processing power for complex database operations.

**4. Operating System:**

* Windows operating system for the database server.

**1.4.3 User Devices:**

**1. Smartphones/Tablets:**

* Compatibility with iOS and Android operating systems.
* Optimization for various screen sizes and resolutions.

**2. Web Browsers:**

* Compatibility with major web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

**1.4.4 Development Environment:**

**1. Programming Languages:**

* Backend: PHP language.
* Frontend: HTML, CSS, JavaScript.

**2. Framework:**

* Web application framework

**3. Database Management System:**

* Choose a suitable DBMS (e.g., MySQL, PostgreSQL, MongoDB) for efficient data storage and retrieval.

**4. APIs:**

* Develop APIs to enable communication between the frontend and backend components.

**5. Integrated Development Environment (IDE):**

* IDEs such as Visual Studio Code, PyCharm, or IntelliJ IDEA for coding and debugging.

**CHAPTER 2**

**FEASIBILITY STUDY**

A Feasibility Study is a crucial step in the initiation phase of a project, providing an in-depth analysis of the proposed system to determine its practicality, viability, and potential success. In the case of a Faculty Contribution Management System, the study would typically include the following components:

* 1. **KEY OBJECTIVES**
     1. **Technical Feasibility**:
* Examine the technical requirements of the proposed system.
* Evaluate the availability of technology and infrastructure needed for implementation.
* Assess the compatibility with existing systems in the educational institution.
  + 1. **Economic Feasibility**:
* Estimate the costs associated with the development, implementation, and maintenance of the system.
* Compare the costs with the expected benefits, including potential savings and increased efficiency.
* Perform a cost-benefit analysis to determine the economic viability of the project.
  + 1. **Legal Feasibility**:
* Identify and analyze any legal or regulatory requirements related to faculty contribution management.
* Ensure that the system complies with data protection and privacy laws.
  + 1. **Operational Feasibility**:
* Analyze how well the proposed system will integrate into daily operations.
* Evaluate the impact on current processes and the ability of faculty members to adapt to the new system.
* Consider the training requirements for staff and faculty.
  + 1. **Scheduling Feasibility**:
* Scheduling a project involves breaking down the tasks and assigning specific timelines for each to ensure the project progresses smoothly.

**2.2 Technical Feasibility**

* Examine the technical requirements of the proposed system.
* Evaluate the availability of technology and infrastructure needed for implementation.
* Assess the compatibility with existing systems in the educational institution.

**2.2.1 Infrastructure Requirements:**

* **Server Infrastructure:** Asses the capacity and scalability of cloud-based servers (e.g., AWS, Azure) to accommodate potential user growth and ensure seamless performance.
* **Database Management**: Evaluate the suitability of database systems (e.g., MySQL, PostgreSQL) for efficient storage and retrieval of user data.

**2.2.2 Software Development:**

* **Programming Languages:** Choose appropriate backend (e.g., PHP) and frontend (e.g., HTML, CSS, JAVASCRIPT) technologies based on developer expertise and project requirements.
* **Framework Selection**: Select a web application framework (e.g., Django, Flask) to streamline development and enhance maintainability.

**2.3 Operational Feasibility**

Operational feasibility is one of the key aspects to consider when evaluating the viability of a project, such as the Faculty Contribution Management System. This feasibility study assesses whether the proposed system can be implemented successfully within the existing organizational structure, processes, and technology.

**2.4 Behavioral Feasibility**

The behavioral feasibility of a Faculty Contribution Management System (FCMS) project refers to the assessment of how well the proposed system aligns with the behavior and attitudes of the individuals who will be using it, specifically the faculty members.

* Evaluate how receptive faculty members are to the idea of a contribution management system. Conduct surveys or interviews to understand their expectations and concerns.
* Consider the ease of use, user interface design, and the overall user experience. A system that is intuitive and user-friendly is more likely to be accepted by faculty members.

**2.5 Schedule Feasibility**

Creating a schedule feasibility for a Faculty Contribution Management System project involves outlining the tasks, estimating the time required for each task, and considering various factors that may impact the project timeline. Below is a simplified example of a project schedule:

**Requirement Analysis:**

* Gather requirements from faculty and administrative staff
* Document functional and non-functional requirements
* Get approval on requirements from stakeholders

**System Design :**

* Create a high-level system architecture
* Design the database schema
* Develop wireframes and mockups for the user interface
* Get design approval from stakeholders

**Development :**

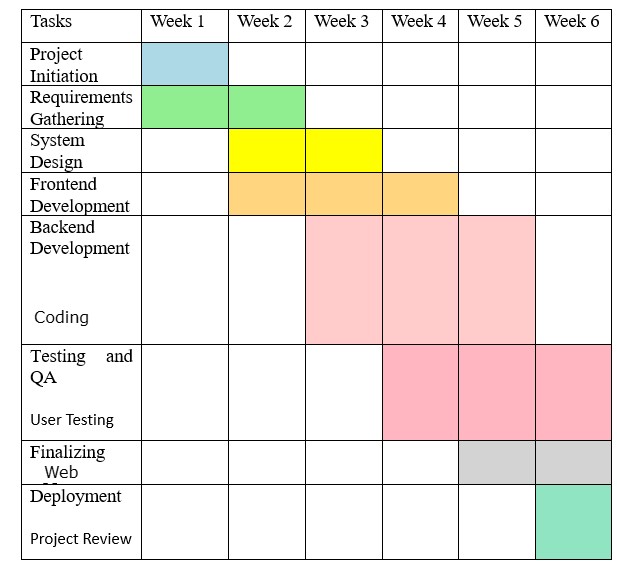
* Implement the backend logic
* Develop the frontend components
* Integrate the database
* Conduct unit testing

**Testing :**

* Conduct system testing
* Perform user acceptance testing (UAT)
* Fix any identified issues

**Deployment:**

* Prepare for deployment
* Roll out the system to users



**Fig. 2.1 Schedule Feasibility (Gantt Chart)**

**CHAPTER 3**

**DATABASE DESIGN**

Designing the database for a Faculty Contribution Management System involves identifying the entities, their attributes, and the relationships between them. Below is a simplified example of how you might approach the database design for such a system:

* 1. **DATABASE TABLES**

Creating a comprehensive database table for the Share Expense app involves

considering the key entities and their attributes. In a simplified example, let's focus on two main entities: Users and Expenses. Here's a basic representation:

1. **Entities:**

* **Faculty Member:** Represents individual faculty members.

Attributes: ID (Primary Key), Name, Email, Department, Position, etc.

* **Contribution:** Represents contributions made by faculty members.

Attributes: ID (Primary Key), Title, Description, Type (e.g., research paper, project), Date, Status, etc.

* **Department:** Represents academic departments within the institution.

Attributes: ID (Primary Key), Name, Head of Department, etc.

* **Admin/User:** Represents administrators or users who manage the system.

Attributes: ID (Primary Key), Username, Password, Email, Role, etc.

1. **Relationships:**

* Each faculty member can submit zero or more contributions.
* Each contribution is submitted by one faculty member.
* Each contribution belongs to one department, and a department can have many contributions.
* Administrators/users have various roles and may have permissions to manage faculty members, contributions, departments, etc.

1. **Database Tables:**

* **Faculty\_Members:**
  + faculty\_id (Primary Key)
  + name
  + email
  + department\_id (Foreign Key)
  + position

...

* **Contributions:**
* contribution\_id (Primary Key)
* faculty\_id (Foreign Key)
* title
* description
* type
* date
* status
* **Departments:**
* department\_id (Primary Key)
* name
* head\_of\_department
* **Admin\_Users:**
  + user\_id (Primary Key)
  + username
  + password
  + email
  + role

1. **Normalization:**

* Ensure that the database is normalized to minimize redundancy and improve data integrity.
* Splitting data into separate tables based on relationships (e.g., separating faculty members from their contributions) helps in achieving normalization.

1. **Indexing:**

* Use indexing on frequently queried columns (e.g., faculty ID, department ID) to improve query performance.

1. **Constraints:**

* Define appropriate constraints (e.g., foreign key constraints, unique constraints) to maintain data integrity.

1. **Security**:

* Implement proper authentication and authorization mechanisms to ensure that only authorized users can access and modify data.

1. **Scalability:**

* Design the database with scalability in mind to accommodate future growth in terms of faculty members, contributions, etc.
  + 1. **Journal Table:**
* **Faculty name :** The name of faculty.
* **Faculty id :** Unique id to identify the faculty.
* **Department :** The department of the faculty.
* **Affiliated :** KIET Affiliation.
* **Publisher :** The name of the publisher.
* **Manuscript :** The manuscript of the Journal.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Faculty name** | **Faculty id** | **Department** | **Affiliated** | **Publisher** | **Manuscript** |
| Mr. Ajay Kumar | 2224mca1145 | CSIT | Yes | Aanand | Live a life |
| Mrs. Surbhi Tyagi | 2224mca1146 | CSE | Yes | Ajanta | Journey |
| Mr. Anil Sharma | 2224mca1147 | MCA | Yes | Aanand | Money |

**Fig. 3.1.1 Journal Table**

* + 1. **Book Table:**
* **Faculty name :** The name of faculty.
* **Faculty id :** Unique id to identify the faculty.
* **Department :** The department of the faculty.
* **Affiliated :** KIET Affiliation.
* **Edition Number :** Edition number of the book.
* **Publisher Type :** Type of the publisher.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Faculty name** | **Faculty id** | **Department** | **Affiliated** | **Edition Number** | **Publisher Type** |
| Mr. Vivek Kumar | 2224mca1156 | CSIT | Yes | 1001 | International |
| Mrs. Mahi Tyagi | 2224mca1157 | CSE | Yes | 1002 | National |
| Mr. Ajay Sharma | 2224mca1158 | MCA | Yes | 1003 | National |

**Fig. 3.1.2 Book Table**

* + 1. **Patent Table:**
* **Appliciant name :** The name of appliciant.
* **Faculty id :** Unique id to identify the faculty.
* **Patent Number :** Patent number of the patent.
* **Status :** Status of the patent.
* **Level :** Level of the patent.
* **SDG :** SDG of the patent.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Appliciant name** | **Faculty id** | **Patent Number** | **Status** | **Level** | **SDG** |
| Mr. Aakash Kumar | 2224mca1165 | CSIT | Published | National | 5 |
| Mrs. Ashu Tyagi | 2224mca1168 | CSE | Granted | National | 7 |
| Mr. Aman Sharma | 2224mca1170 | MCA | Granted | International | 9 |

**Fig. 3.1.3 Patent Table**

* + 1. **FDP Table:**
* **Faculty name :** The name of faculty.
* **FDP id :** Unique id to identify the FDP.
* **FDP Name :** The name of the FDP.
* **Week :** Duration of the FDP.
* **Mode :** Mode of the FDP.
* **Sponsered by :** Sponser of the FDP.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Faculty name** | **FDP id** | **FDP Name** | **Week** | **Mode** | **Sponsered by** |
| Mr. Ajay Kumar | 2224mca1145 | CSIT | Yes | Aanand | Live a life |
| Mrs. Surbhi Tyagi | 2224mca1146 | CSE | Yes | Ajanta | Journey |
| Mr. Anil Sharma | 2224mca1147 | MCA | Yes | Aanand | Money |

**Fig. 3.1.4 FDP Table**

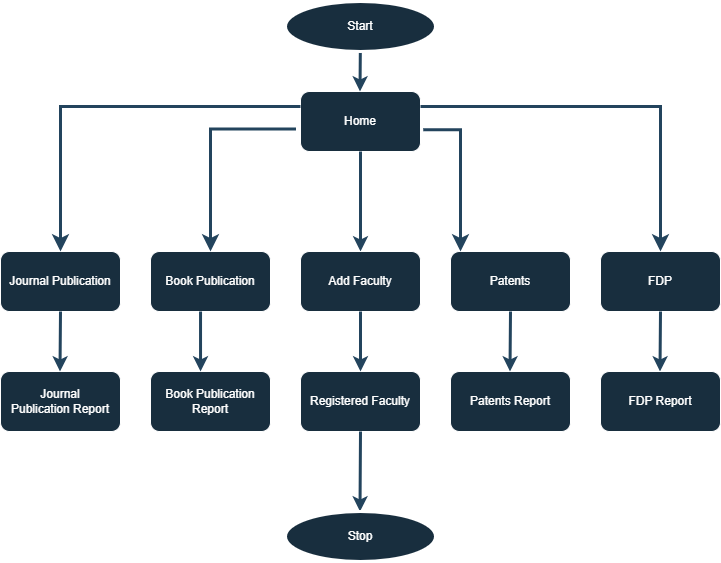
* + 1. **Faculty Table:**
* **Faculty name :** The name of faculty
* **Faculty id :** Unique id to identify the faculty.
* **Department :** The department of the faculty.
* **Designation :** The designation of the faculty.
* **Email id :** Email id of the faculty.
* **Phone:** Phone number of the faculty.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Faculty name** | **Faculty id** | **Department** | **Designation** | **Email id** | **Phone** |
| Mr. Ajay Kumar | 2224mca1145 | CSIT | Professor | ajay@gmail.com | Live a life |
| Mrs. Surbhi Tyagi | 2224mca1146 | CSE | Professor | surbhi@gmail.com | Journey |
| Mr. Anil Sharma | 2224mca1147 | MCA | Professor | anil@gmail.com | Money |

**Fig. 3.1.5 Faculty Table**

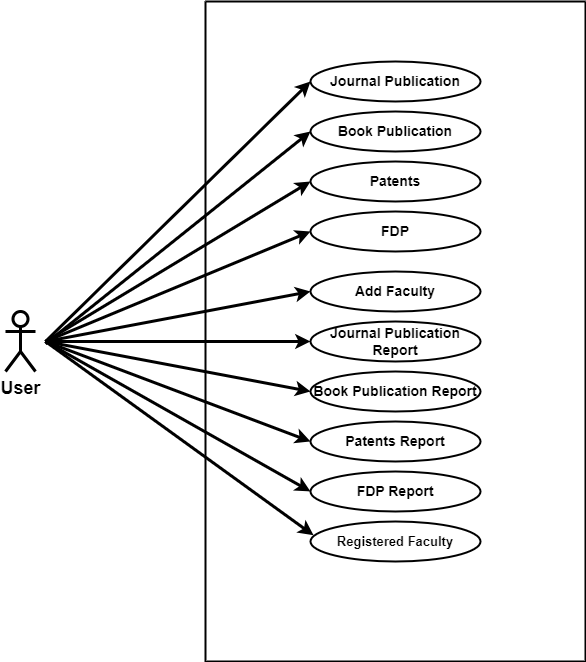
* 1. **FLOWCHART**

Creating a flowchart for a Faculty Contribution Management System involves breaking down the system into various processes and illustrating the flow of information or actions between them. Below is a simplified example of a flowchart for such a system.



**Fig. 3.3 Flowchart Diagram**

* 1. **USE CASE DIAGRAM**

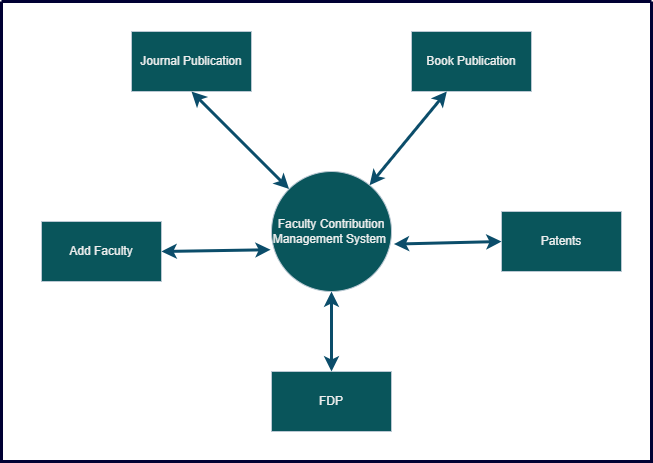
A Use Case Diagram is a visual representation in Unified Modeling Language (UML) that illustrates the interactions between different actors (users or external systems) and a system, showcasing various use cases and their relationships. In the context of a Faculty Contribution Management System project, the Use Case Diagram would outline the functionalities and interactions between different entities involved in the system. 

**Fig. 3.4 Use Case Diagram**

* 1. **DATA FLOW DIAGRAM**

**3.4.1 Faculty Contribution Management System** **DFD Level 0**

This is the Zero Level DFD of Faculty Management System, where we have eloborated the high level process of Faculty Management. It's a basic overview of the whole Faculty Management System or process being analyzed or modeled. It's designed to be an at-a-glance view of Teacher, Course and Subject showing the system as a single high-level process, with its relationship to external entities of Leave, Faculty and Student. It should be easily understood by a wide audience, including Leave, Student and Teacher in zero leve DFD of Faculty Management System, we have described the high level flow of the Faculty Management system.

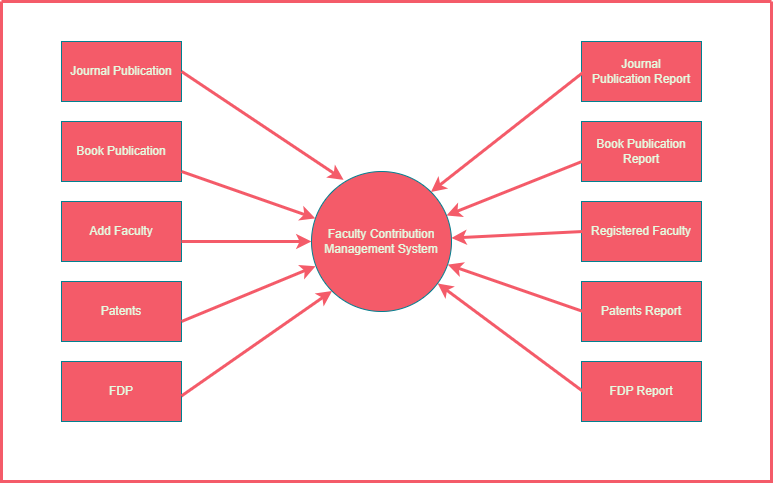


**Fig. 3.6 Level 0 DFD**

* Managing all the Faculty
* Managing all the Journal Publication
* Managing all the Book Publication
* Managing all the Patents
* Managing all the Faculty Development Program
* Managing all the Report

**3.4.2 Faculty Contribution Management System** **DFD Level 1**

First Level DFD (1st Level) of Faculty Management System shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Faculty Management System system as a whole. It also identifies internal data stores of Subject, Course, Teacher, Timecard, Student that must be present in order for the Faculty Management system to do its job, and shows the flow of data between the various parts of Leave, Student, Course, Subject, Teacher of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD. You will highlight the main functionalities of Faculty Management.

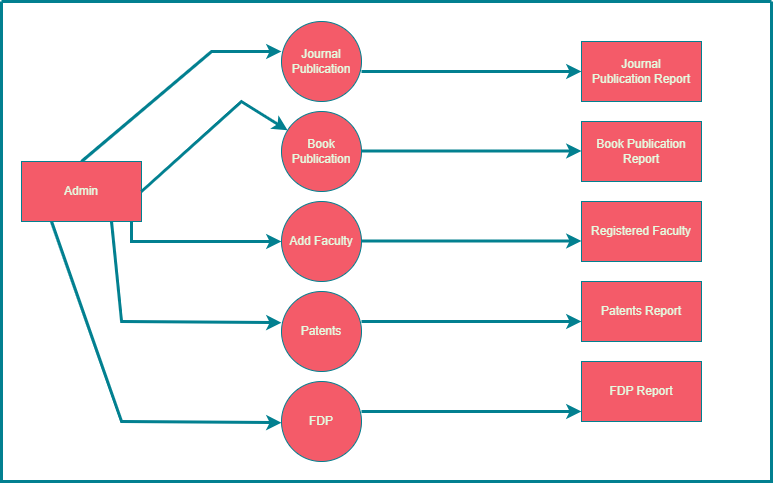


**Fig. 3.7 Level 1 DFD**

* Processing Faculty records and generate report of all Faculty
* Processing Journal records and generate report of all Journal
* Processing Book records and generate report of all Book
* Processing Patent records and generate report of all Patent
* Processing FDP records and generate report of all FDP

**3.4.3** **Faculty Contribution Management System** **Level 2**

DFD Level 2 then goes one step deeper into parts of Level 1 of Faculty Management. It may require more functionalities of Faculty Management to reach the necessary level of detail about the Faculty Management functioning. First Level DFD (1st Level) of Faculty Management System shows how the system is divided into sub-systems (processes). The 2nd Level DFD contains more details of Subject, Course, Teacher, Timecard, Student, Faculty, Leave.



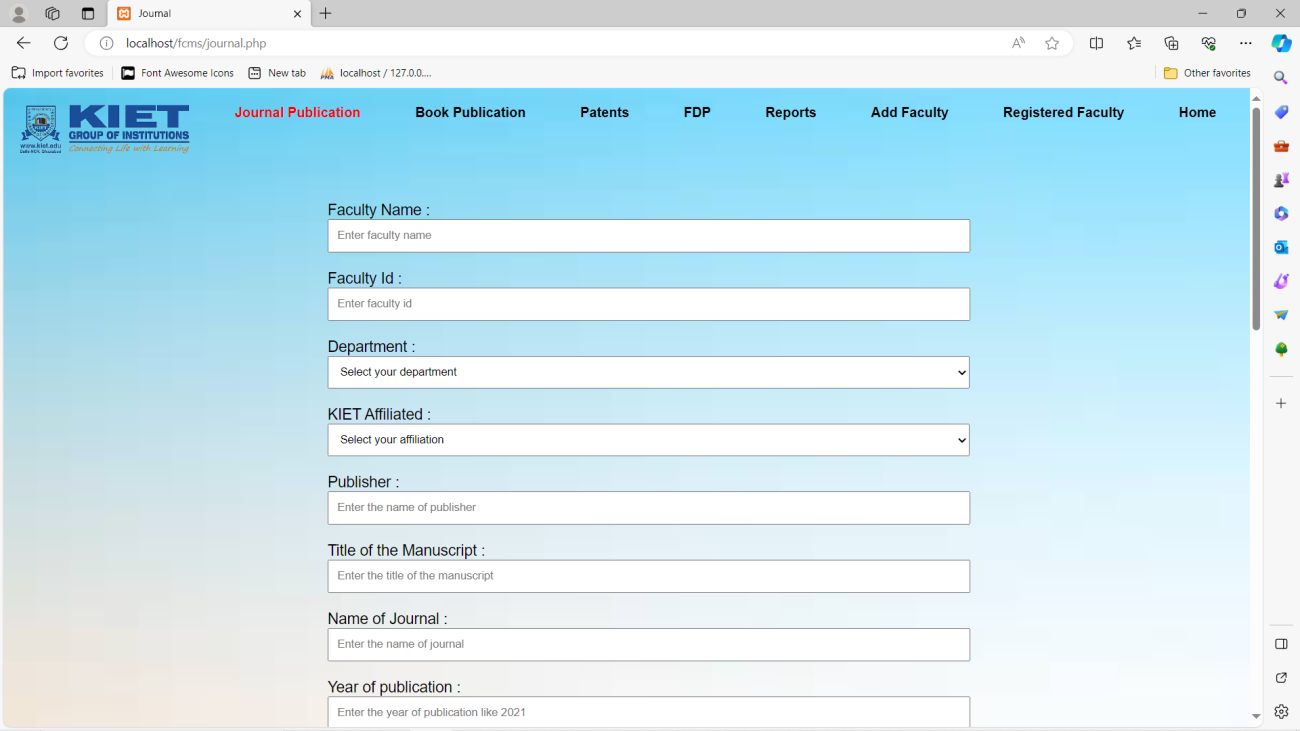
**Fig. 3.8 Level 2 DFD**

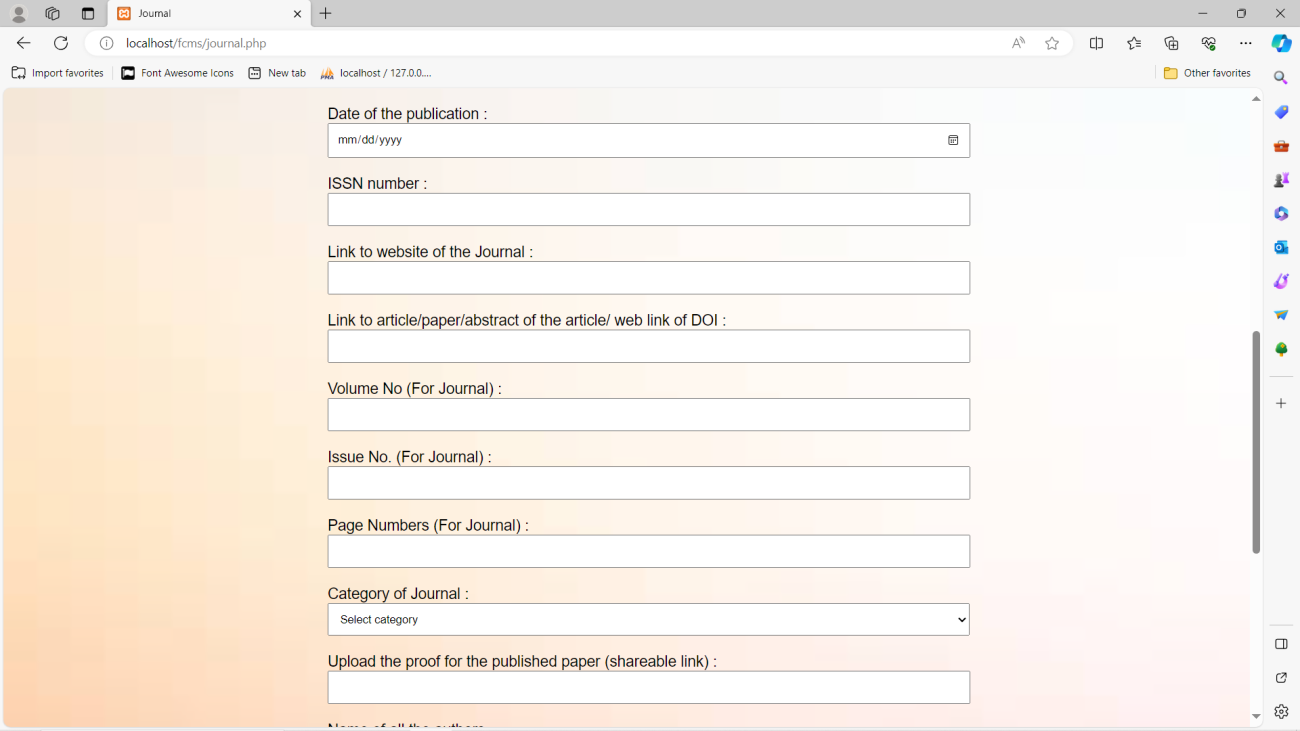
* Admin can add, edit, delete and view the records of Journals, Books, Patents, FDP
* Admin can manage all the details of Faculty
* Admin can also generate reports of Journals, Books, Patents, FDP
* Admin can apply different level of filters on report of Journals, Books, Patents, FDP
* Admin can tracks the detailed information of Faculty, Journals, Books, Patents, FDP

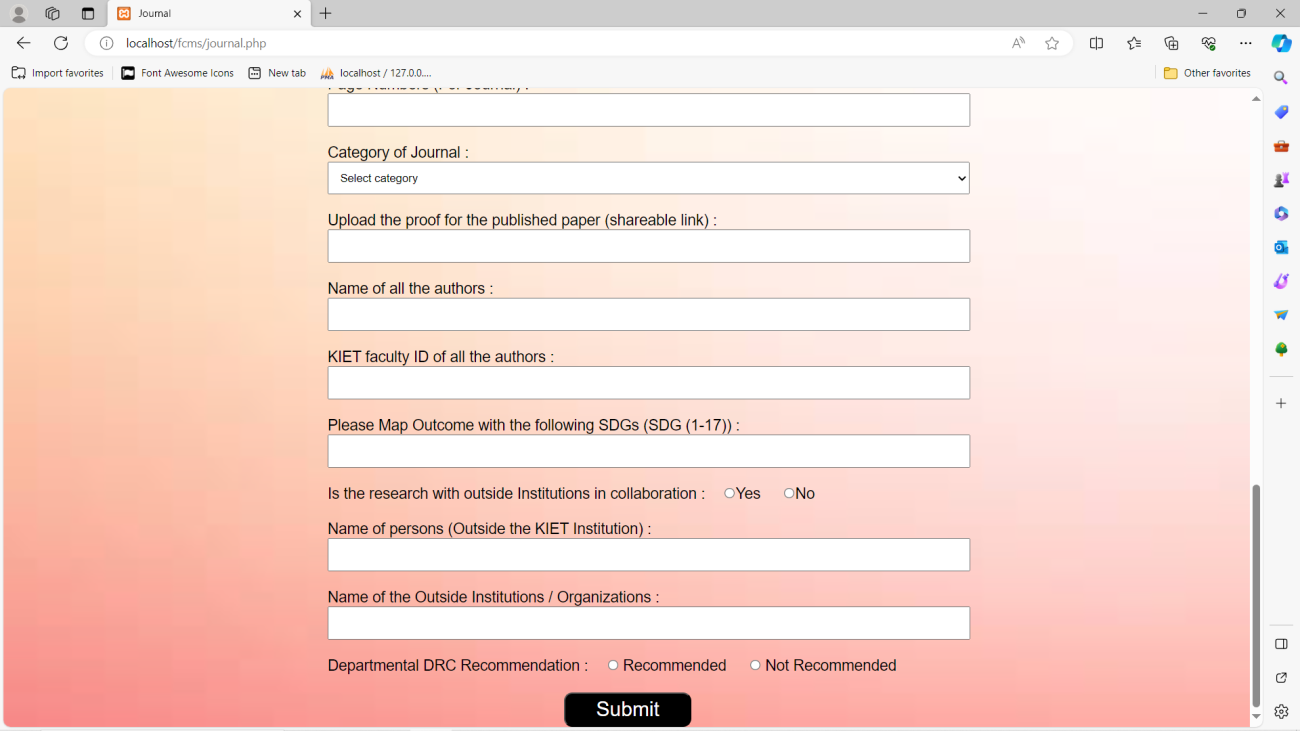
**CHAPTER 4**

**PROJECT PROCESS**

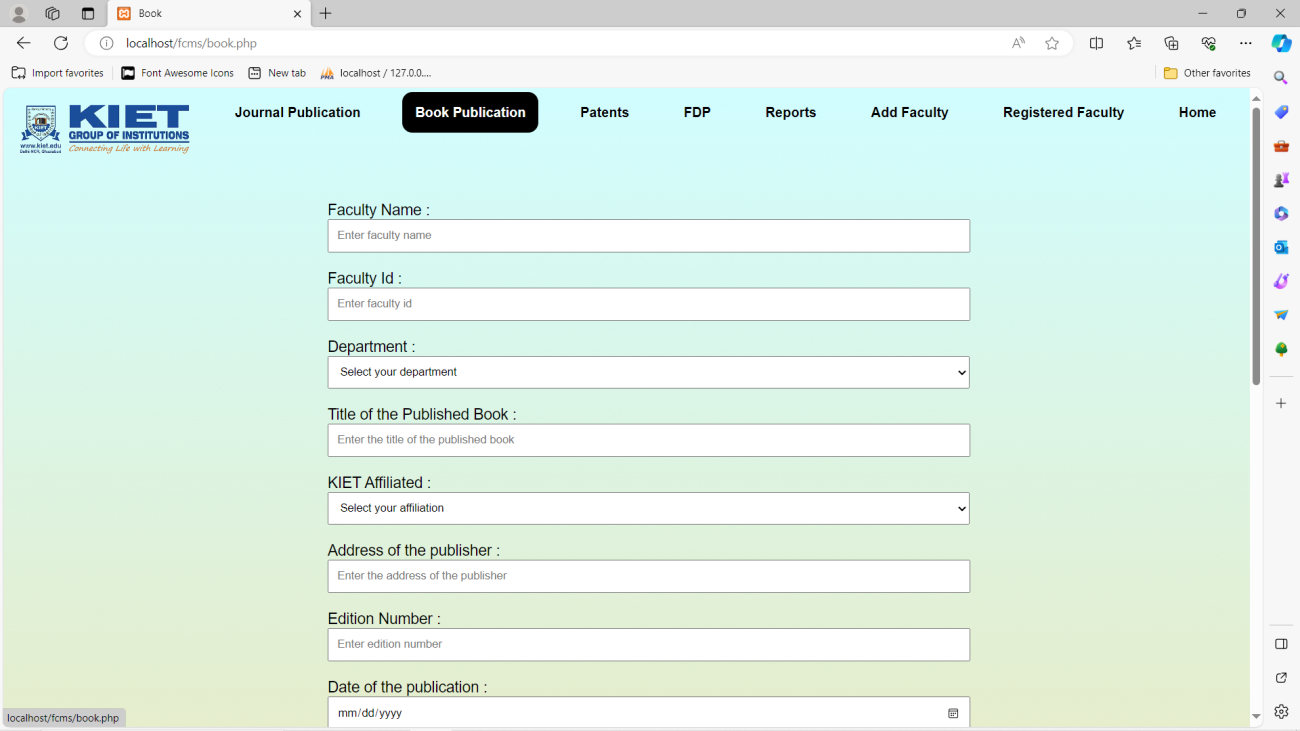
The Faculty Contribution Management System is a mini project designed to streamline and enhance the management of academic contributions by faculty members in an educational institution. The system provides a centralized platform for faculty members to submit, track, and manage their contributions such as research papers, projects, publications, and other scholarly activities. It also offers administrative functionalities for authorized personnel to oversee and generate reports on faculty contributions.

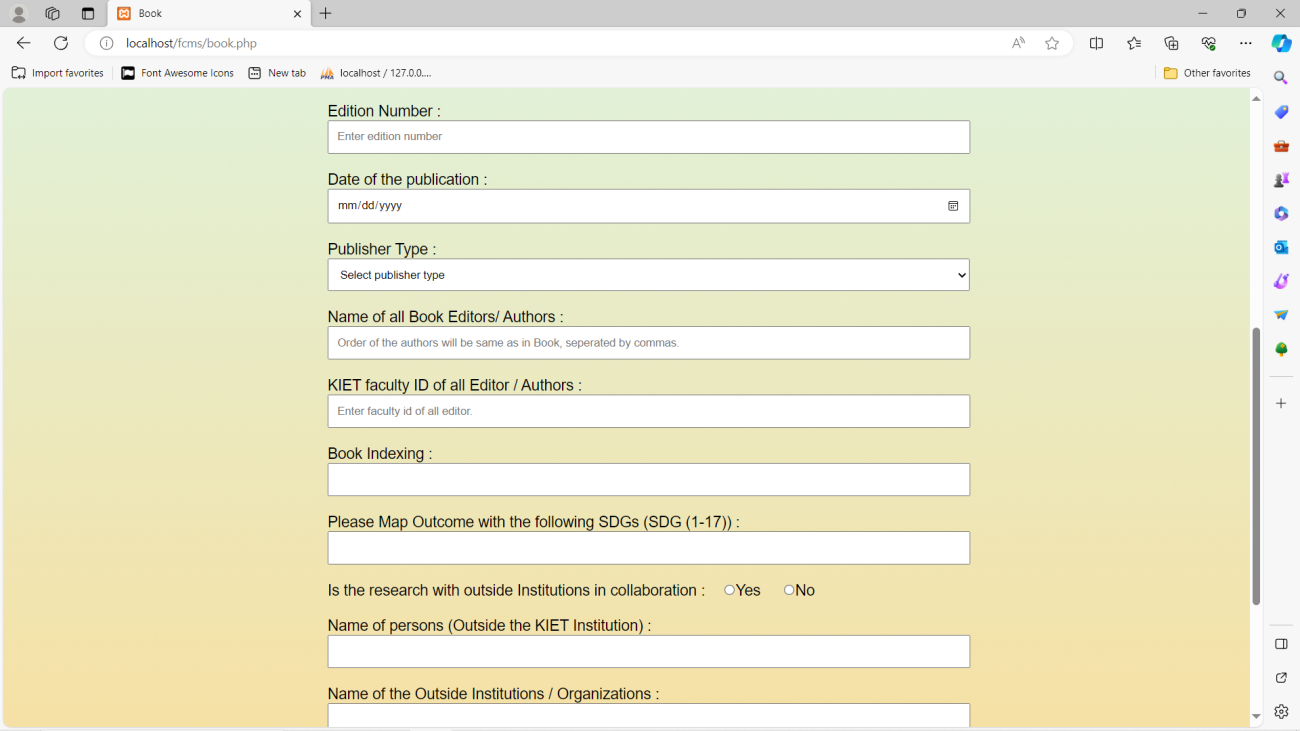


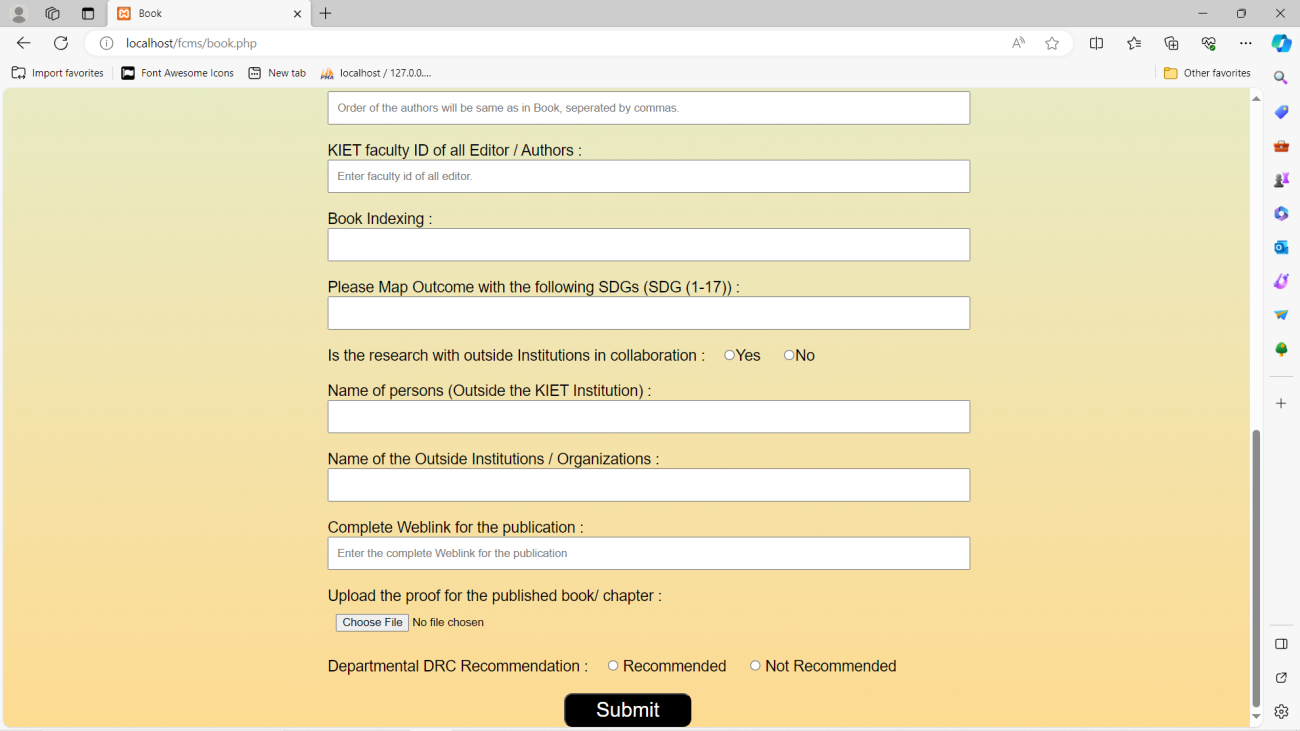




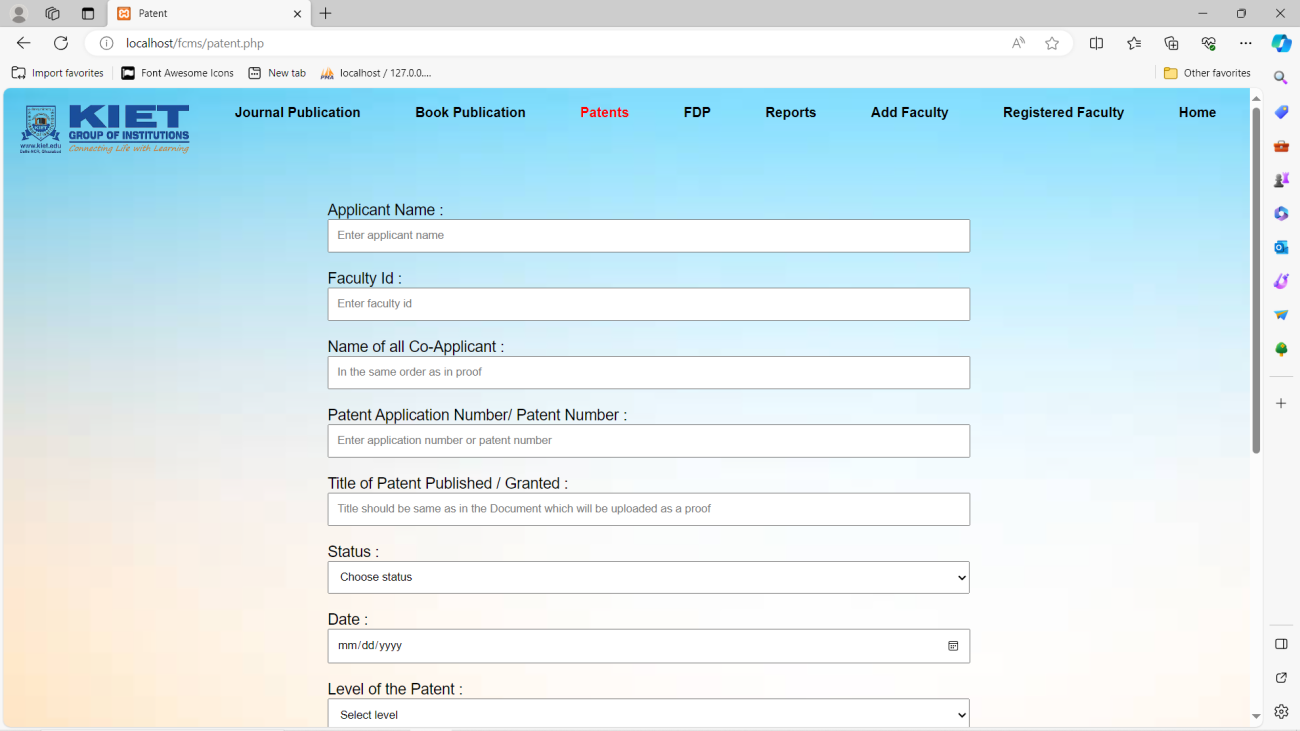
**Fig. 4.1 Journal Publication**

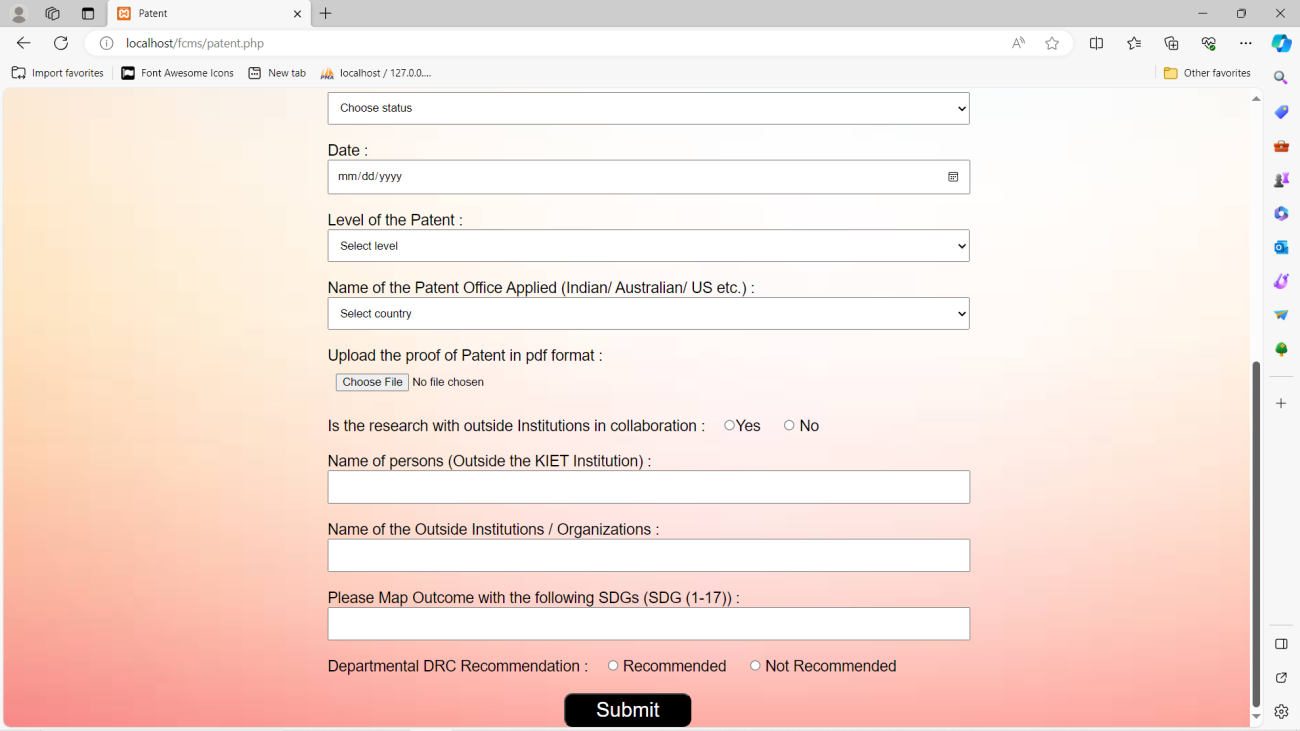




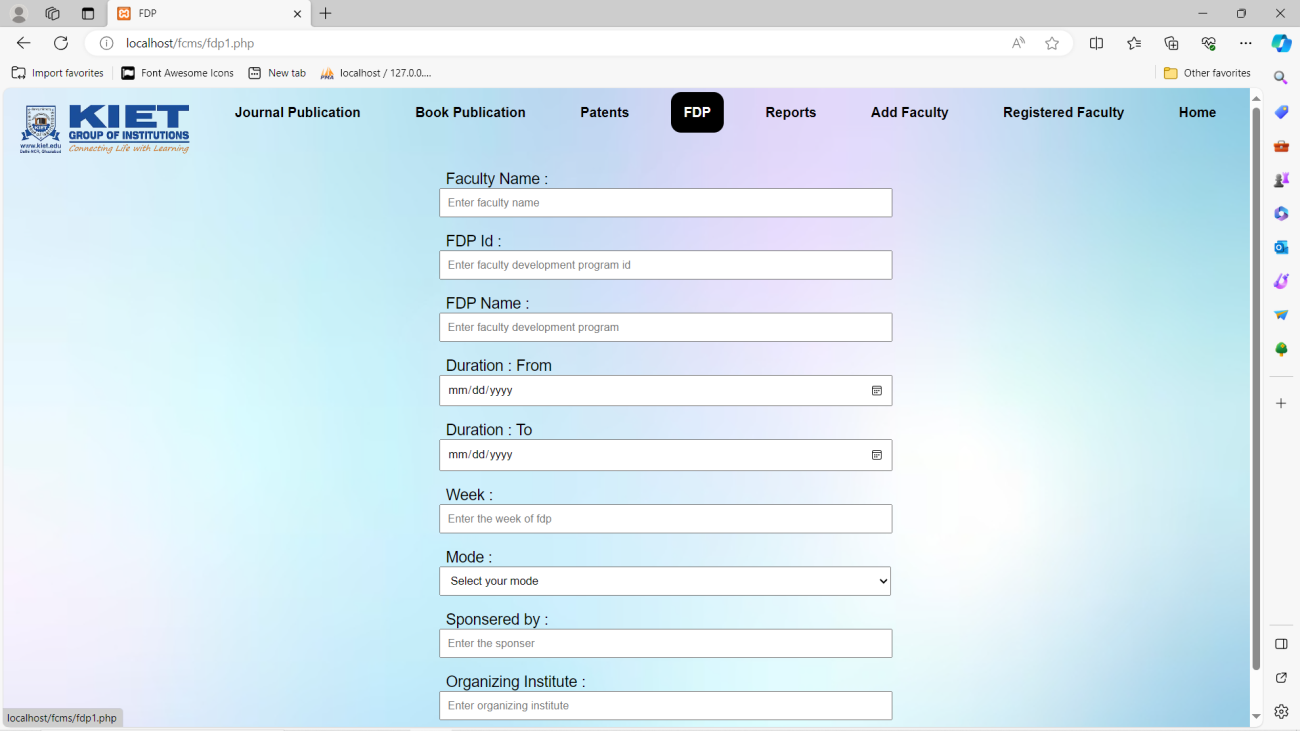


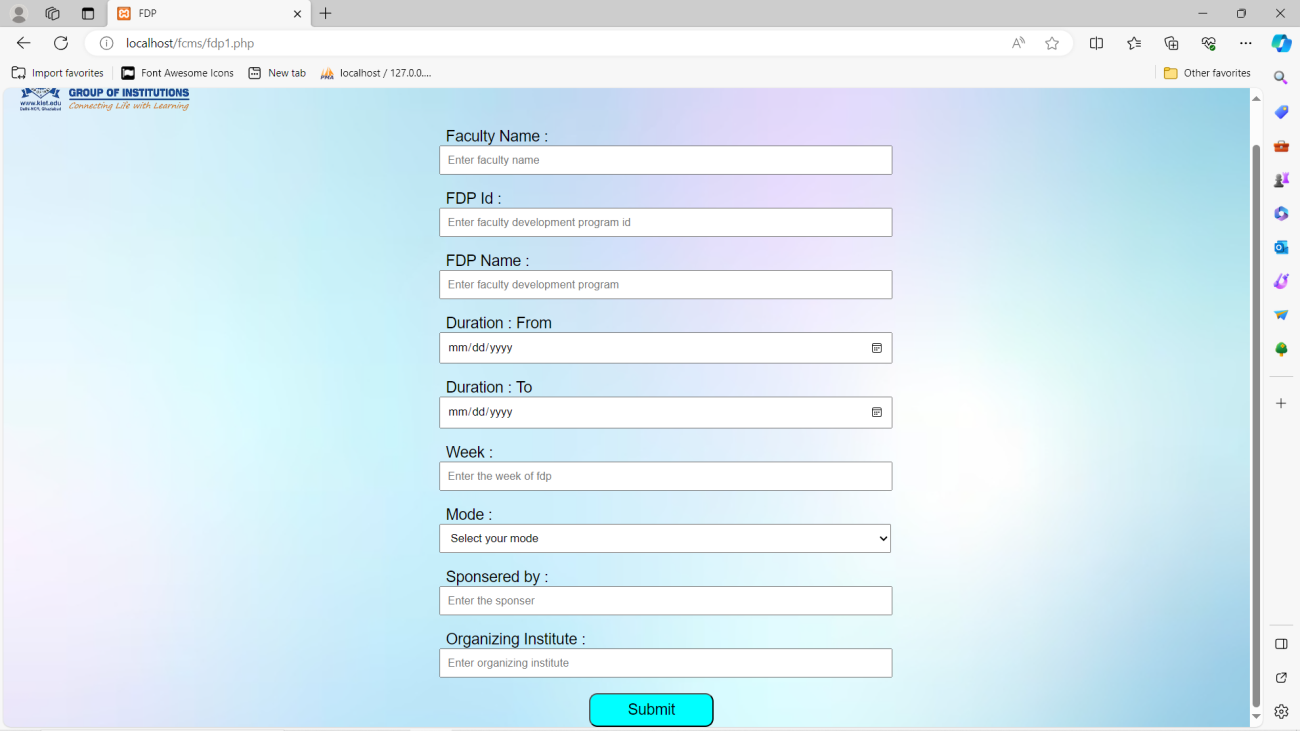
**Fig. 4.2 Book Publication**

****

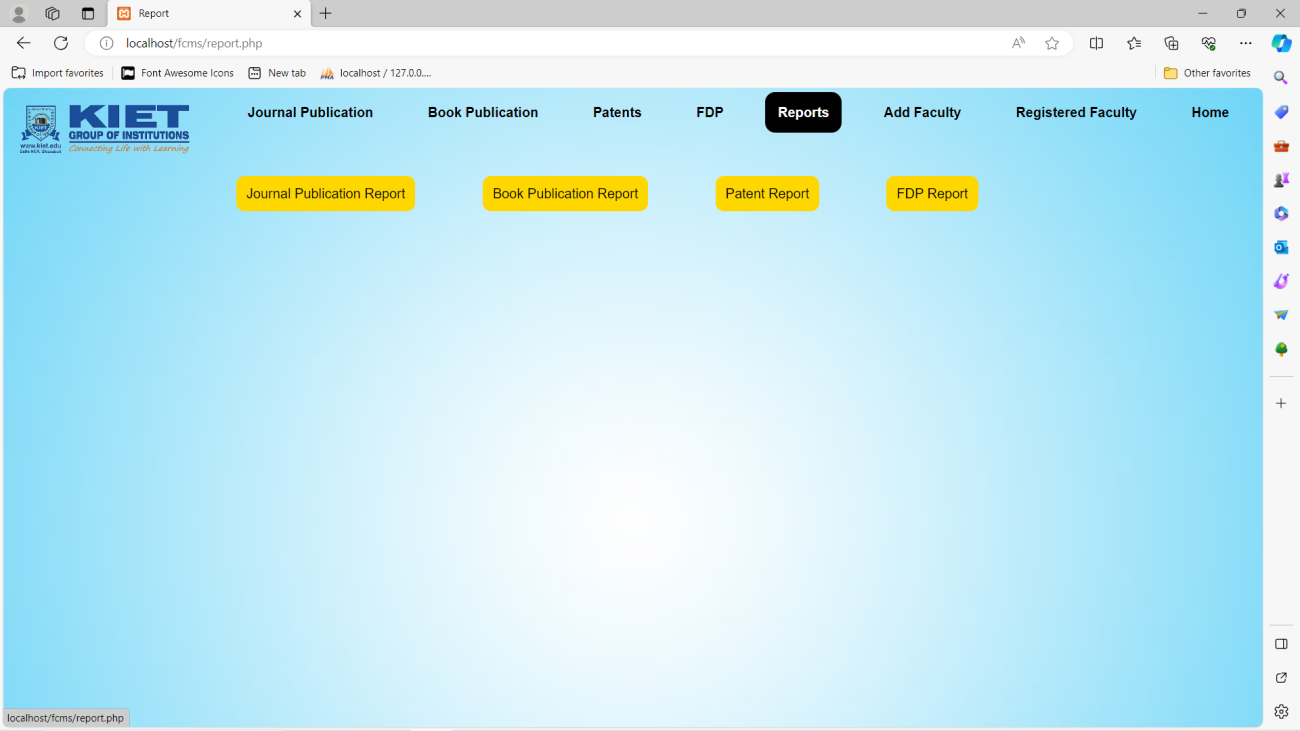
****

**Fig. 4.3: Patents**

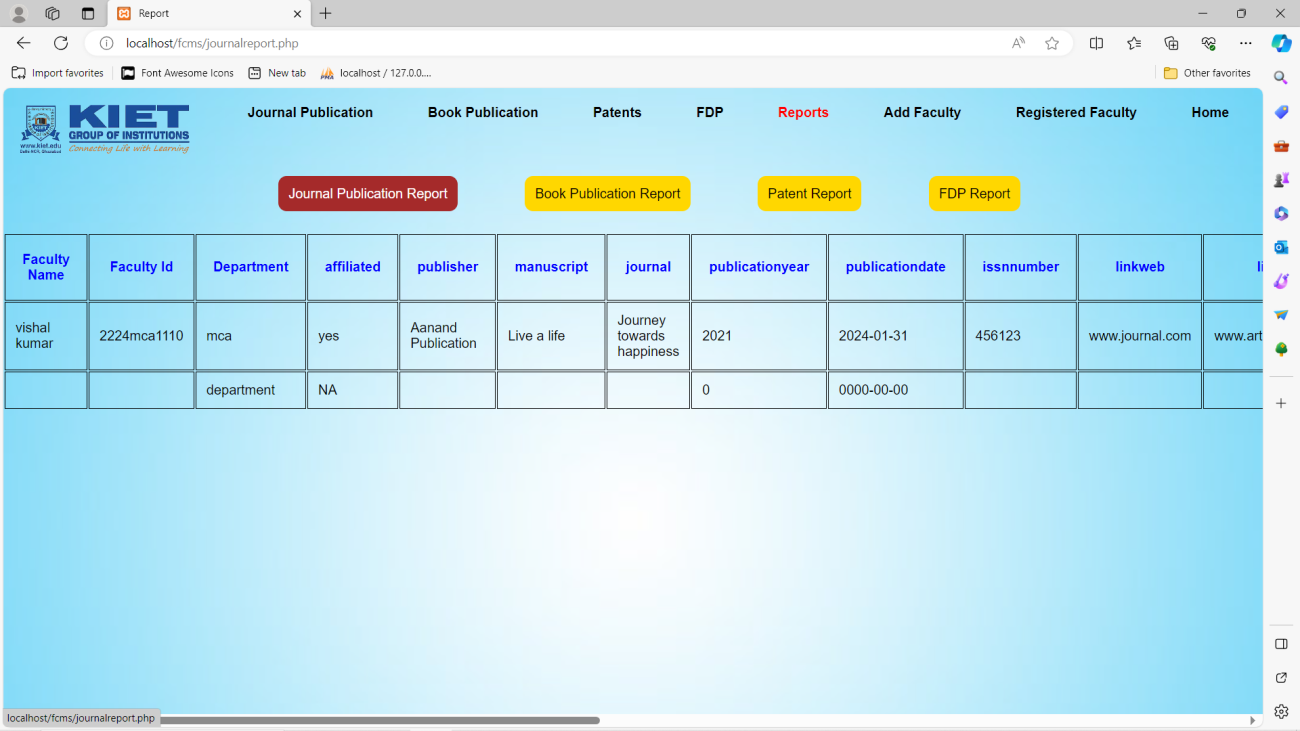




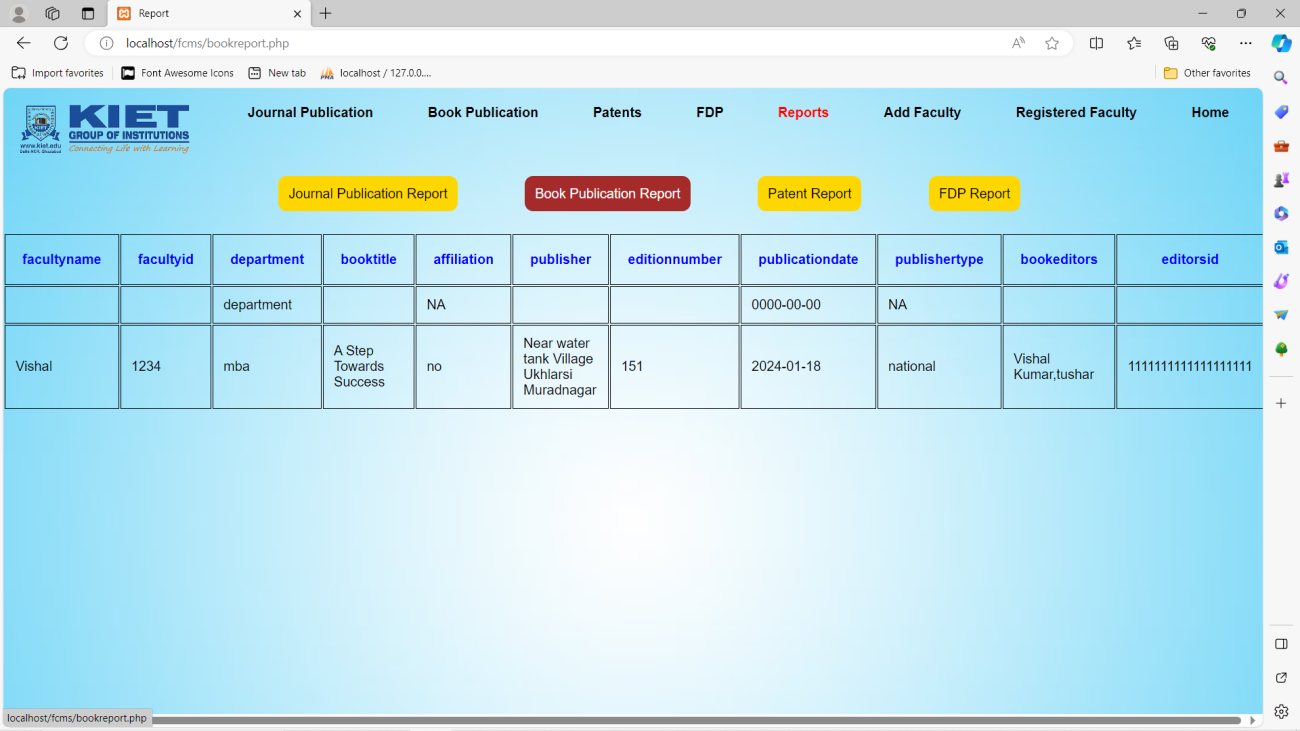
**Fig. 4.4: FDP**



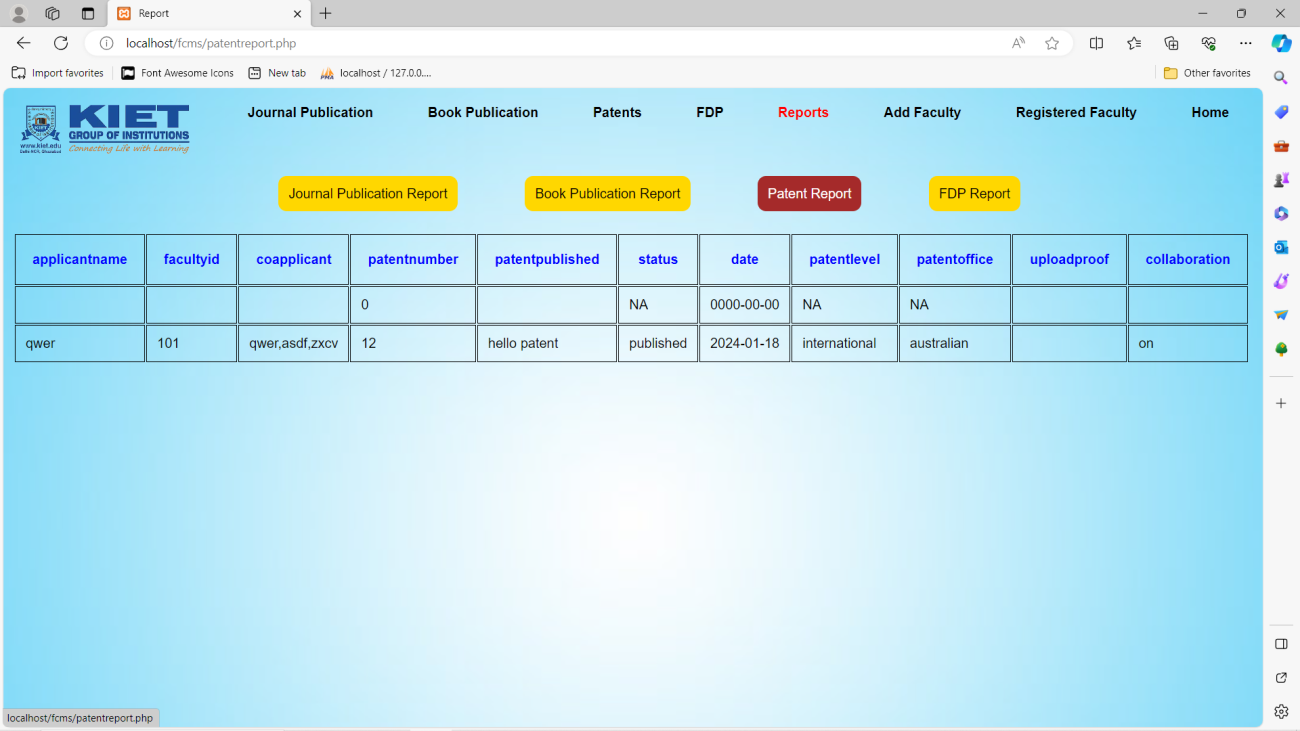
**Fig. 4.5: Reports**



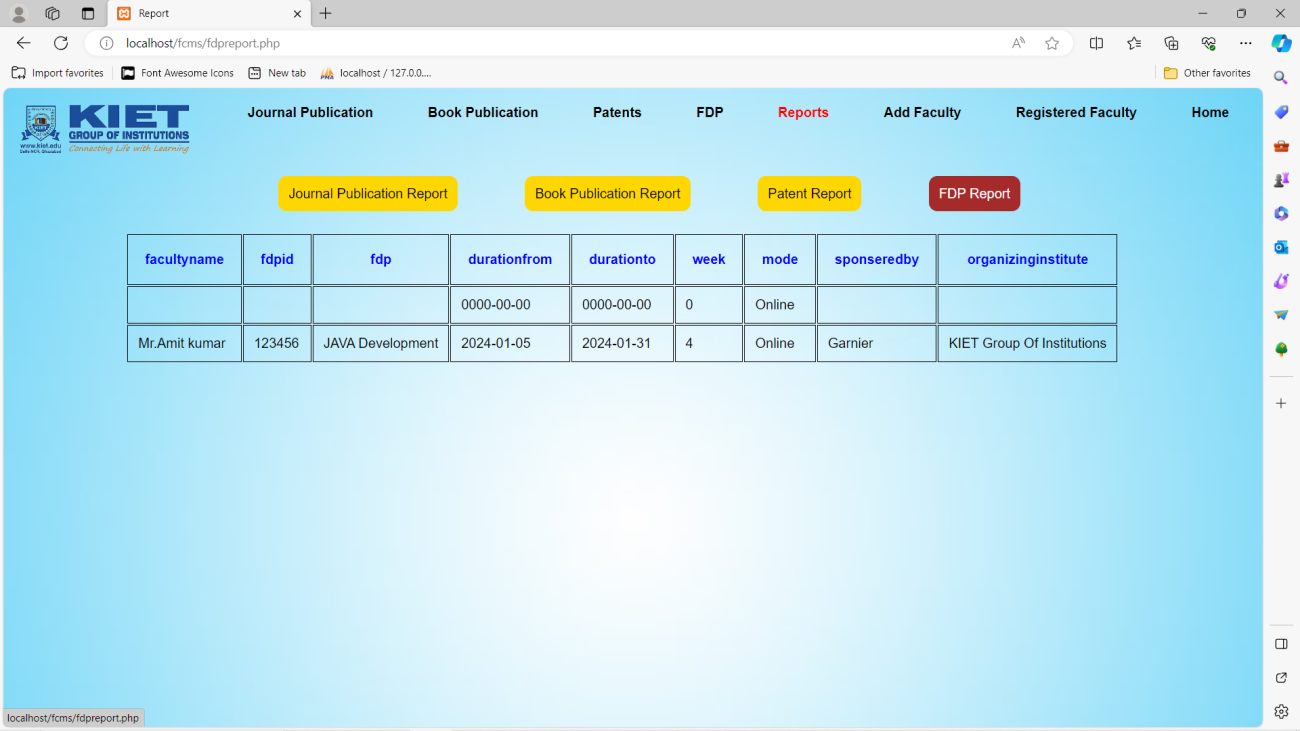
**Fig. 4.6: Journal Publication Report**



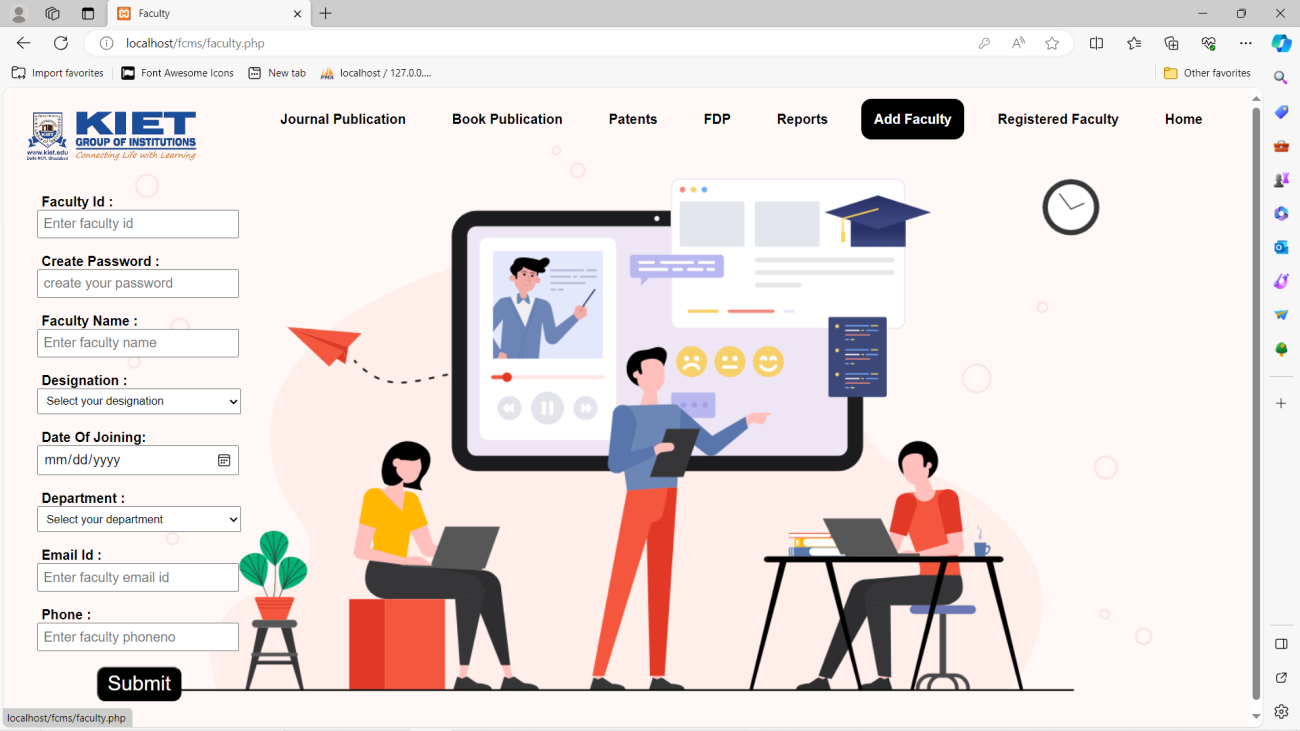
**Fig. 4.7: Book Publication Report**

****

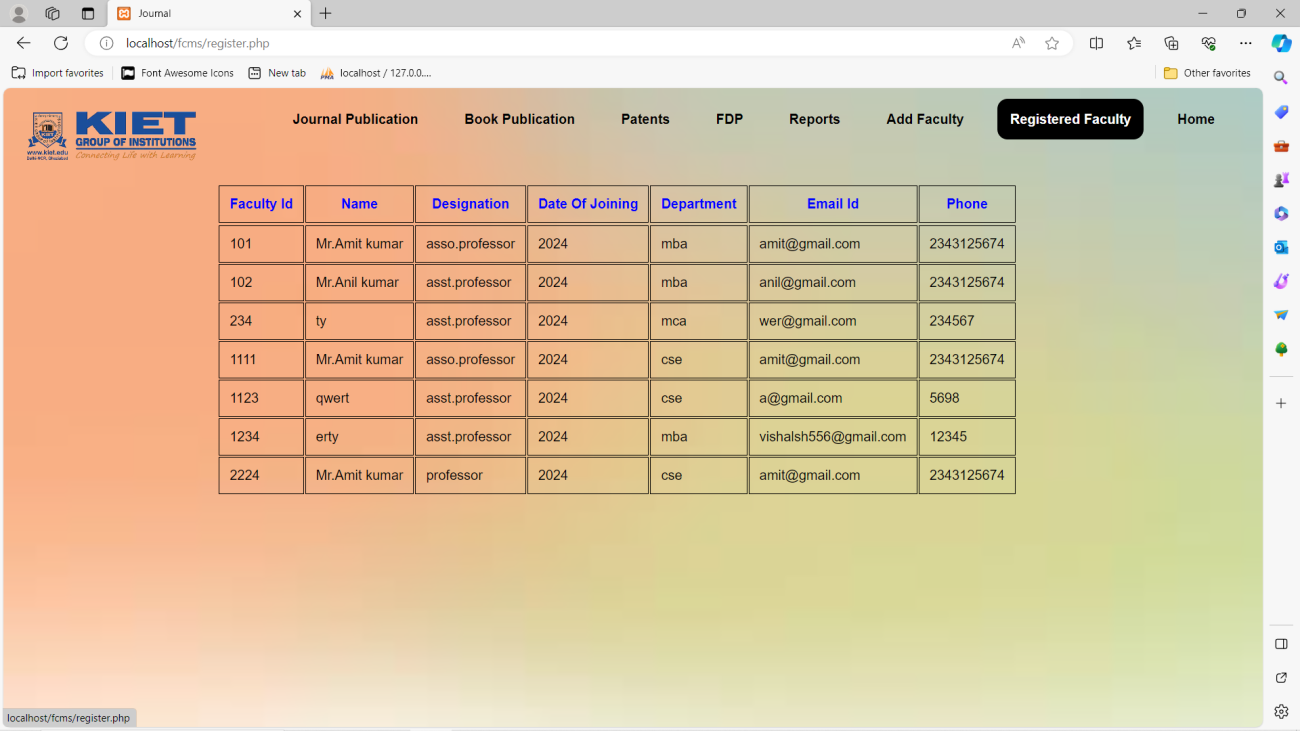
**Fig. 4.8: Patent Report**



**Fig. 4.9: FDP Report**

****

**Fig. 4.10: Add Faculty**

****

**Fig. 4.11: Registered Faculty**

**CHAPTER 5**

**TESTING**

Testing is a crucial phase in the development of any software system, including the Faculty Contribution Management System. It ensures that the system functions correctly, meets requirements, and is free from defects. Here are some key aspects and types of testing that could be applied to a Faculty Contribution Management System:

**5.1 Key Objectives:**

The key objective of a Faculty Contribution Management System is to streamline and manage the various contributions and activities of faculty members within an educational institution. This system aims to enhance efficiency, transparency, and accountability in tracking and evaluating faculty contributions. Here are the key objectives of such a system:

**5.1.1 Functional Testing:**

* **Contribution Submission:**Test the process of submitting contributions to ensure it works as intended.
* **View and Edit Contributions:** Ensure faculty members can view and edit their contributions.
* **Admin Functions:** Test administrative functions like generating reports, managing user roles, and overseeing the system.

**5.1.2 Integration Testing:**

* **Verify Interaction between Modules:** Ensure that different modules of the system, such as user authentication, contribution submission, and reporting, work seamlessly together.
* **Data Integration:** Check if data flows smoothly between the database and various components.

**5.1.3 Performance Testing:**

* **Scalability:** Evaluate how the system performs with varying numbers of faculty members and contributions.
* **Response Time:** Measure the response time for critical functions, such as submission and report generation, to ensure they meet acceptable standards.

**5.1.4 Security Testing:**

* **Data Security:** Verify that sensitive data (such as personal information and contributions) is stored securely and access is appropriately restricted.
* **Authentication and Authorization:** Ensure that only authorized users can access specific functionalities.

**5.1.5 Usability Testing:**

* **User Interface (UI) Testing:** Evaluate the user interface for intuitiveness, consistency, and ease of use.
* **Accessibility:** Ensure that the system is accessible to users with different abilities.

**5.1.6 Regression Testing:**

* **Ensure Changes Do Not Break Existing Functionality:** After any updates or modifications, test to ensure that existing features still work as expected.

**5.1.7 User Acceptance Testing (UAT):**

* **Involve End Users:** Allow faculty members and administrators to test the system to ensure it meets their needs and expectations.

**5.1.8 Load Testing:**

* **Simulate Concurrent Users:** Test how the system performs under heavy loads, simulating multiple users accessing and interacting with the system simultaneously.

**5.1.9 Compatibility Testing:**

* **Browser Compatibility:** Ensure the system works seamlessly on different web browsers.
* **Device Compatibility:** Check compatibility with various devices (desktops, laptops, tablets, etc.).

**5.1.10 Data Integrity Testing:**

* **Ensure Data Accuracy:** Confirm that the data stored in the system is accurate and reliable.

**5.2 Testing Methodologies:**

**5.2.1 Unit Testing:**

Assess individual code units to ensure they perform as expected in isolation.

**5.2.2 Integration Testing:**

Evaluate the interactions between different components and modules to ensure seamless integration.

**5.2.3 System Testing:**

Verify the system as a whole to ensure that all components work together cohesively.

**5.2.4 Acceptance Testing:**

Validate that the Faculty Contribution Management System Project meets the specified requirements and gains approval from stakeholders.

**5.3 Collaborative Testing:**

The testing phase involves collaboration among developers, quality assurance professionals, and end-users. Communication is crucial to understanding requirements, addressing issues, and refining the system iteratively.

**BIBLIOGRAPHY**

Here's a hypothetical bibliography for a Faculty Contribution Management System:

1. **Author(s):** Smith, John; Patel, Priya

* **Title:** "Design and Implementation of a Faculty Contribution Management System: A Case Study in Higher Education Institutions"
* **Publication Source:** Journal of Educational Technology and Management
* **Year:** 2020

1. **Author(s):** Kumar, Ravi; Singh, Ananya

* **Title:** "Faculty Contribution Management Systems: Challenges and Opportunities in Indian Universities"
* **Publication Source:** Proceedings of the International Conference on Information Systems (ICIS)
* **Year:** 2018

1. **Author(s):** Gupta, Aakash; Sharma, Meera

* **Title:** "Enhancing Faculty Productivity through Contribution Management Systems: Lessons from Implementations in Engineering Colleges"
* **Publication Source:** International Journal of Educational Technology
* **Year:** 2019

1. **Author(s):** Khan, Ali; Rahman, Sameer

* **Title:** "Faculty Contribution Tracking Systems: A Comparative Study of Open Source Solutions"
* **Publication Source:** Journal of Information Systems and Technology
* **Year:** 2021

1. **Author(s):** Das, Swati; Chatterjee, Rajesh

* **Title:** "Implementing Faculty Contribution Management Systems in Indian Institutes: Success Factors and Challenges"
* **Publication Source:** Proceedings of the National Conference on Educational Technology
* **Year:** 2017

1. **Author(s):** Mishra, Neha; Reddy, Arjun

* **Title:** "Evaluation of Faculty Contribution Management Systems: A Case Study of Universities in South India"
* **Publication Source:** International Journal of Higher Education Management
* **Year:** 2022

1. **Author(s):** Gupta, Sanjay; Verma, Nidhi

* **Title:** "Faculty Contribution Management Systems: Best Practices and Future Directions"
* **Publication Source:** Conference on Educational Technology Innovations
* **Year:** 2019

1. **Author(s):** Sharma, Rohit; Jain, Deepika

* **Title:** "Adoption of Faculty Contribution Management Systems: A Study of Factors Influencing User Acceptance"
* **Publication Source:** Journal of Educational Administration and Management
* **Year:** 2020

1. **Author(s):** Patel, Dipesh; Desai, Ritu

* **Title:** "Faculty Contribution Management Systems in Indian Universities: A Review of Literature and Research Directions"
* **Publication Source:** International Journal of Information Sciences
* **Year:** 2018

1. **Author(s):** Singh, Vikram; Gupta, Pooja

* **Title:** "Data Analytics in Faculty Contribution Management Systems: Leveraging Insights for Institutional Improvement"
* **Publication Source:** Journal of Data Science and Analytics in Education
* **Year:** 2021

**REFERENCE**

1. W3Schools.

https://www.w3schools.com/html/

1. Bootstrap Documentation.

https://getbootstrap.com/

1. Chatgpt

[https://chat.openai.com/](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)

1. Font Awesome Documentation.

[https://fontawesome.com/](https://fontawesome.com/v5.15/how-to-use/on-the-web/setup/getting-started)

1. Stack overflow

https://stackoverflow.com/