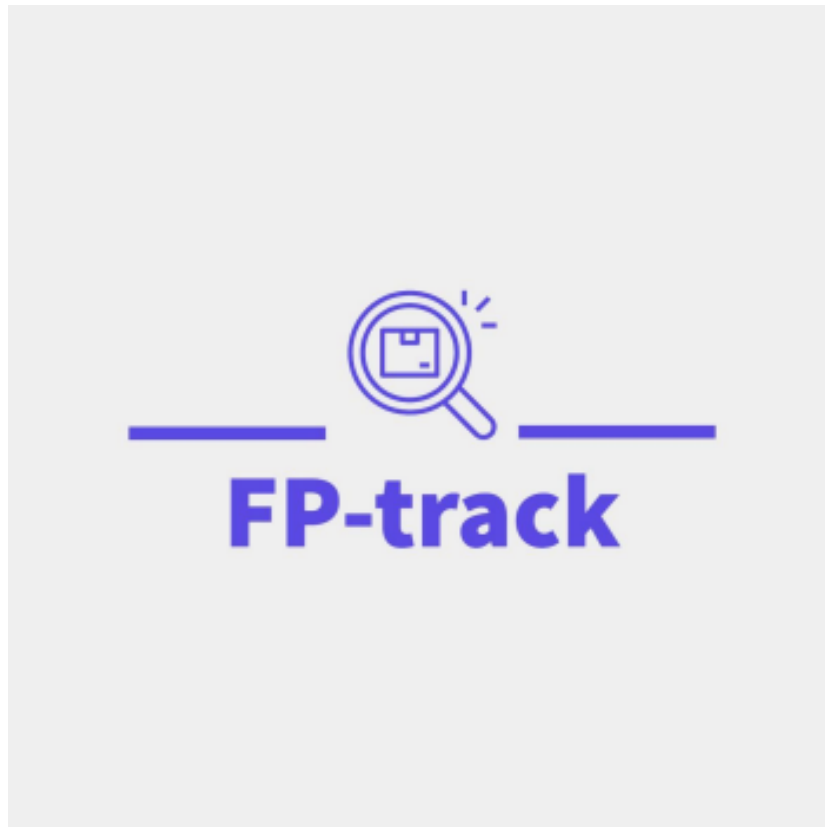


- ☒ i. ~~Prepare the detailed problem statement~~
- ☒ ii. ~~Prepare the software requirement (functional, non-functional) specification for your project.~~
- ☐ iii. Explore different kinds of design for the interface and choose the optimal design in terms of user friendliness, like easy navigation, minimal clicks, neat design, and consistent layouts across all web pages that belong to the web application.
- ☒ iv. ~~Identify the functionalities – say Login & Registration Module, Item Module, Payment Module, Logout Module~~
- ☐ v. Design and Develop appropriate the logic based on the study objectives
- ☒ vi. ~~Implement and integrate module using modern JavaScript frameworks– NodeJS for Server and other client of your choice such as ReactJS, ExpressJS etc.~~
- ☒ vii. ~~Use the new features of the tool for better functionalities~~
- ☐ viii. Document the features or techniques that were not used in your project and justify why it was not used. Suggest how it can be used in the future work
- ☒ ix. ~~Submit a detailed report of your project~~

Funded Project and Resource Tracking System



Project Design and Specifications Document

Version 2.0

Funded Project and Resource Tracking System	1
Problem Statement	7
Proposed Solution	7
Updating Project Details on the Web-Application	7
Internal Projects	7
External Projects	8
Resource Management	8
Report Generation	8
Entity Description	9
User Stories	11
Open Access (Unauthenticated Guest User)	11
Admin	11
Applicant - Faculty/Student	11
Resource Manager	11
Software Requirements Specification	12
Introduction	12
Updating Project Details on the Web-Application	12
Internal Projects	12
External Projects	13
Resource Management	13
Report Generation	13
Entity Description	13
Purpose	14
Scope	14
Definitions, Acronyms, and Abbreviations	15
References	15
Overview	15
Overall Description	17
Product Perspective	17
Product Functions	18
User characteristics	19
Constraints	19
Assumptions and Dependencies	19
Specific Requirements	20
Functional	20
Updating Project Details on the Web-Application	20
Internal Projects Management	20

External Projects Management	20
Resource Management	20
Report Generation	21
Login Capabilities	21
Mobile Devices	21
Alerts	21
Non-Functional	21
Usability	21
Reliability	21
Availability	22
Accuracy	22
Access Reliability	22
Resource Utilization	22
Supportability	22
Internet Protocols	22
Information Security Requirement	22
Maintenance	22
Standards	22
Performance	22
Design Constraints	23
Software Language Used	23
Development Tools	23
Class Libraries	23
Online User Documentation and Help System Requirements	23
Interfaces	23
User Interfaces	24
Hardware Interfaces	24
Software Interfaces	24
Communications Interfaces	24
Licensing Requirements	24
Legal, Copyright, and Other Notices	24
Applicable Standards	24
Supporting Information	24
Design Strategies Adopted (refer to this)	25
Modules Implemented	26
Identification of Modules	26
Conceptual Real-World Entity Category List	26
Identification of Real-World Entities	26
Elimination of Trivial Entities	27
Final List of Entities — Modeled as Modules	27

Categorization of Modules	28
Identification of Associations between Modules	29
Association Category list	29
Associations Identified and Categorization	29
Definition of Associations and their notations	31
Generalization	31
Aggregation	31
Composition	31
Multiplicity Based Associations	32
Domain View of Modules	32
Elaborate View of Modules	33
Logic Representation & Workflow Description	34
Identified Workflows and System Logic	34
Main Workflow 1: Student/Faculty Registers Project	34
Main Workflow 2: Request for Resources	34
Alternate Workflow 1: Student/Faculty Updates Project Progress	34
Alternate Workflow 2: Student/Faculty Records Project Completion	35
Alternate Workflow 3: Resource Manager Reclaim Resources	35
Alternate Workflow 4: Resource Manager Reports Faulty Components	35
Alternate Workflow 5: Guest Generates/Views Reports	36
Alternate Workflow 6: Guest Searches for Projects	36
Identification of Generic System States in the Workflow	36
Workflows — Flowchart Representation	38
Workflow 1: Record Project Application	38
Workflow 2: Resource Allocation Request	39
Workflow 3: Project Milestones and Completion Updation	40
Workflow 4: Resource Management	41
Workflow 5: Project Search and Report Generation	41
Components of the Integrated Implementation	42
Backend — NodeJS with Express	42
Frontend — ReactJS with Next	44
Remote Database — MongoDB	46
Users	46
Proposals	46
Projects	47
Resource Groups	47
Resources	47
Resource Assignments	48
End-User Interface	49
User Dashboard	49

Submit Proposals	49
View Proposals	50
View Projects - Overview	50
View Projects - Detailed	52
Approve/Reject Proposals [Admin]	54
Allocate Resources for Project [Resource Manager]	55
User Management [Admin]	55
Future Improvements (refer to this)	56
Features and Techniques not Implemented	56
Suggested Improvements	56

Problem Statement

Every year, several SSNites face issues in submitting and receiving timely responses on their funded project proposals. There is no centralized system to apply, approve and track the progress of applications.

Furthermore, there is a need to track resources available with the department and repurpose them for upcoming projects with similar requirements, as opposed to granting funds to acquire redundant components.

Proposed Solution

We propose a centralized online system to manage internally and externally funded projects in the college. The following aspects will be handled by the system:

Updating Project Details on the Web-Application

The web-application can be used to update and track internal funding requests. In addition, it will serve as a repository to record the status of externally funded projects affiliated to the college.

Internal Projects

Prior to approving funds and allocating resources, the internal funding committee can find resources already available with the department, allocate them to the project and grant funds only for components that aren't already available with the department.

Faculty and students applying for internal funding must update details and status of their applications, detailing aspects about the funding authority, domain of work, budget requested, etc.

Approved projects can then be used by the applicants to record regular progress, directly visible to the research cell. The final outcomes of the projects will be added to the project in the end and archived into the database.

Intermediate and final outcomes of the projects must also be updated by the applicants, and will be archived after completion.

Upon project completion, the newly acquired resources are inventorized on the system and submitted to the department.

External Projects

Applicants applying for external funding must update details and status of their applications, detailing aspects about the funding authority, domain of work, budget requested, etc.

Intermediate and final outcomes of the projects must also be updated by the applicants, and will be archived after completion.

Resource Management

The system also maintains a repository of project resources already available with the system. This includes the availability status of the resources.

The department can choose to allocate these resources to newly approved applicants, whilst approving funds for other requirements alone. This ensures efficient usage and economic repurposing of the college's resources.

Faculty/Students, upon project completion, will deposit the newly acquired components with the department. These items will be inventoried on the system for reuse in future.

Report Generation

Summary statistics of applied and approved funding projects can be curated through the system based on multiple filtering criteria.

This can prove useful during research showcases and academic year progress presentations. Furthermore, it can be pivotal in auditing research and academic activities.

Entity Description

Project types can be one of the following types:

- Externally funded projects
 - Faculty projects
- Internally funded projects
 - Internally Funded Faculty Projects
 - Internally Funded Student Projects

Proposal/Applications can be set to one of the following statuses:

- Applied

- Shortlisted for Presentation
- Approved
- Rejected

Approved projects are maintained as a separate repository with the following possible statuses:

- Ongoing
- Completed

Regular updates and final outcomes are recorded along with the completed projects before archival. Outcomes will be represented as:

- Publications
- Link to publications project OR DBLP/GScholar page
- Patents
- Forwarded to External Funding

Resource repository to inventory components already available, categorized as:

- Resources currently under use
- Available resources
- Faulty resources

Report generation with filters, such as:

- Calendar period
- Project members
- Project domain
- Budget
- Status

Status updation interface will contain:

- Status upgrades
- Email status triggers

User Stories

Open Access (Unauthenticated Guest User)

- View statistics
- Generate reports and graphs
- View approved projects list

Admin

- User and Database administration
- Status updation of applicants

Applicant - Faculty/Student

- Updates information about funding applications on the portal
- Updates progress of projects on the web-application
- Records final outcome of projects

Resource Manager

- Updates status and inventories resources when they are surrendered
- Monitors resource inventory managed on the website

Software Requirements Specification

Introduction

Research and development projects are commonly characterized by funding applications to support the project at its multiple stages. Such funding agencies exist within the university and as part of governmental organizations and other venture and incubation organizations.

Every year, several SSNites face issues in submitting and receiving timely responses on their funded project proposals. There is no centralized system to apply, approve and track the progress of applications.

Furthermore, in the case of funding within the university department, there is a need to track resources available with the department and repurpose them for upcoming projects with similar requirements, as opposed to granting funds to acquire redundant components.

We propose a centralized online system to manage internally and externally funded projects in the college. The following aspects will be handled by the system:

Updating Project Details on the Web-Application

The web application can be used to update and track internal funding requests. In addition, it will serve as a repository to record the status of externally funded projects affiliated with the college.

Internal Projects

Before approving funds and allocating resources, the internal funding committee can find resources already available with the department, allocate them to the project, and grant funds only for components that aren't already available with the department.

Faculty and students applying for internal funding must update details and status of their applications, detailing aspects about the funding authority, domain of work, the budget requested, etc. Approved projects can then be used by the applicants to record regular progress, directly visible to the research cell.

The final outcomes of the projects will be added to the project in the end and archived into the database. Intermediate and final outcomes of the projects must also be updated by the applicants and will be archived after completion. Upon project completion, the newly acquired resources are inventoried on the system and submitted to the department.

External Projects

Applicants applying for external funding must update details and status of their applications, detailing aspects about the funding authority, domain of work, the budget requested, etc. Intermediate and final outcomes of the projects must also be updated by the applicants and will be archived after completion.

Resource Management

The system also maintains a repository of project resources already available with the system. This includes the availability status of the resources. The department can choose to allocate these resources to newly approved applicants, whilst approving funds for other requirements alone. This ensures efficient usage and economic repurposing of the college's resources.

Faculty/Students, upon project completion, will deposit the newly acquired components with the department. These items will be inventoried on the system for reuse in the future.

Report Generation

Summary statistics of applied and approved funding projects can be curated through the system based on multiple filtering criteria.

Entity Description

1. Project types can be one of:
 - Externally funded projects
 - Faculty projects
 - Internally funded projects
 - Internally Funded Faculty Projects
 - Internally Funded Student Projects
2. Proposal/Applications can be set to one of the following statuses:
 - Applied
 - Shortlisted for Presentation
 - Approved
 - Rejected
3. Approved projects are maintained as a separate repository with the following possible statuses:
 - Ongoing
 - Completed
4. Regular updates and outcomes are recorded along with the completed projects before archival.
5. Outcomes will be represented as:

- Publications
 - Link to publications project OR DBLP/GScholar page
 - Patents
 - Forwarded to External Funding
6. Resource repository to inventory components already available, categorized as:
- Resources currently underuse
 - Available resources
7. Report generation with filters, such as:
- Calendar period
 - Project
 - Members
 - Project
 - Domain
 - Budget
 - Status
8. Status updation interface

To record and display the project updates through a centralized portal, that can be monitored by project mentors and the research cell as well.

Purpose

The purpose of the **Software Requirements Specification (SRS)** document is to describe the external behavior of the Funded Project Tracking System proposed here. This specification defines and describes the operations, interfaces, performance, and quality assurance requirements of the Funded Project Tracker. The document also describes the non-functional requirements such as the user interfaces. It also describes the design constraints that are to be considered when the system is to be designed, and other factors necessary to provide a complete and comprehensive description of the requirements for the software. The Software Requirements Specification (SRS) captures the complete software requirements for the system, or a portion of the system.

Scope

The Online Funded Project Tracker that is to be developed exposes a centralized online system for faculty and students to manage internally and externally funded projects in the college. The following aspects and features will be handled by the system:

- An *unauthenticated guest* user can:
 - view funding statistics
 - generate reports and graphs of applicant-wise stats
 - view approved projects list

- An *admin* user can:
 - perform user and database administration
 - update status of applicants
- An *applicant faculty/student* user can:
 - update information about funding applications on the web-application
 - update progress of projects on the web-application Records final outcome of projects
- A *resource manager* user can:
 - update status and inventories of resources when they are surrendered
 - monitor resource inventory stored on the web-application

The features that are described in this document are used in the future phases of the software development cycle. The features described here meet the needs of all the users. The success criteria for the system is based on the level up to which the features described in this document are implemented in the system.

Definitions, Acronyms, and Abbreviations

- SSN - Sri Sivasubramaniya Nadar College of Engineering
- CSE - Computer Science and Engineering
- IFSP - Internally Funded Student Project
- IFFP - Internal Funded Faculty Project
- FPT - Funded Project Tracker

References

This SRS document uses the following documents as references:

Problem Statement Specification: To specify the existing system, its pitfalls and the proposed solution along with entity descriptions and user stories are presented in the problem statement document stored here: <https://www.github.com/karthik-d/Funded-Project-Tracker>

Current State of Funded Project Listings: SSN's official website features a basic tabular display of all funded project details. This listing is basic in its functionality and does not allow resource reallocation. This webpage can be found here:

<https://www.ssn.edu.in/college-of-engineering/it-newsletter-faculty-internal-funded-projects/>

Overview

This SRS will provide a detailed description of the FPT System. This document will provide the outline of the requirements, an overview of the characteristics and constraints of the system.

The next section of the SRS will provide the general factors that affect the product and its requirements. It provides the background for those requirements. The items such as product perspective, product function, user characteristics, constraints, assumptions and dependencies, and requirements subsets are described in this section.

The final section of SRS contains all the software requirements mentioned in section 2 in detail sufficient enough to enable designers to design the system to satisfy the requirements and testers to test if the system satisfies those requirements.

Overall Description

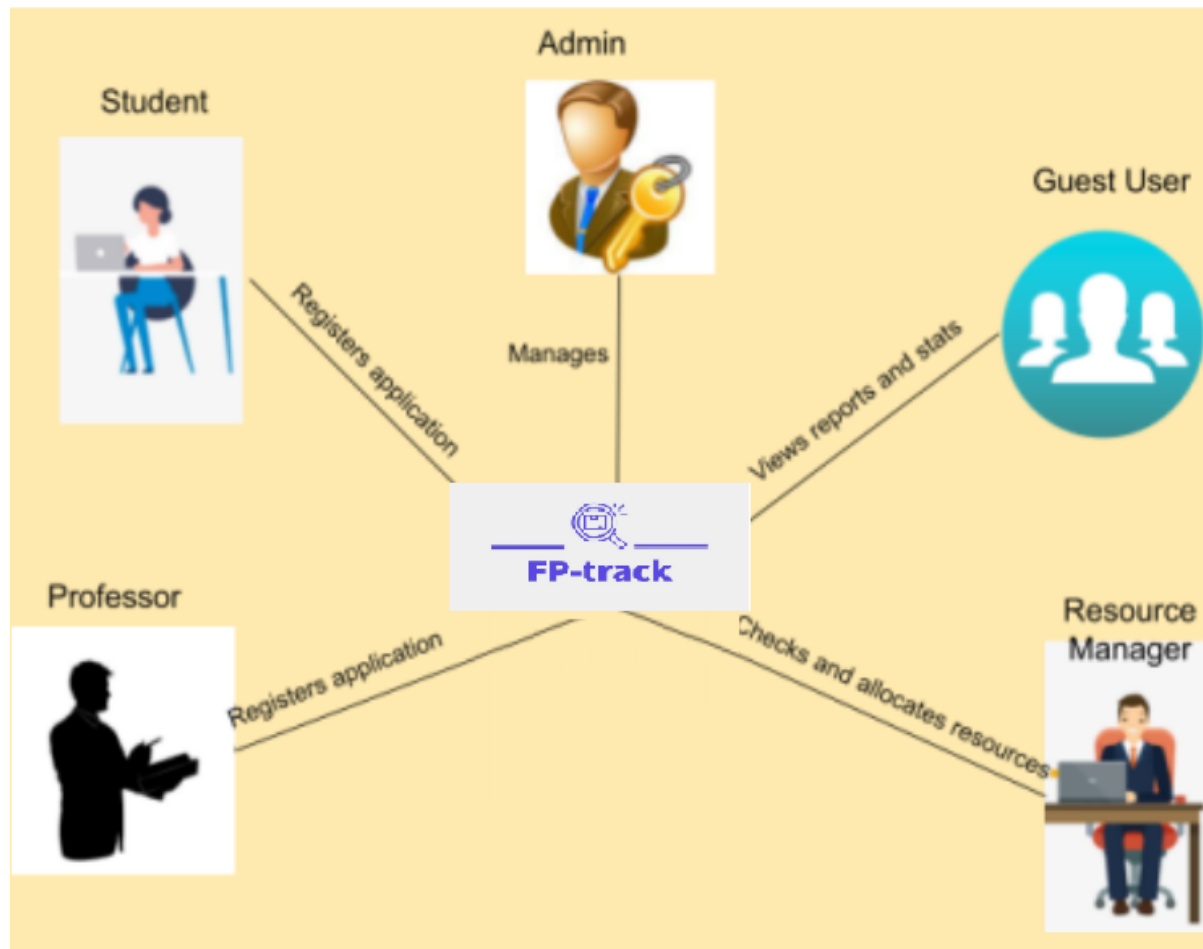
Product Perspective

The project tracking system is a package that will be used by students as well as faculty members to upload their project ideas and get the appropriate resources required for the project. The system developed will be used by the SSN College of Engineering to manage the Internally funded projects.

The product to be developed has interactions with the following users:

- Student applying for funding
- Faculty applying for funding
- Resource manager
- Admin
- Guest user

The complete overview of the system is as shown in the overview diagram below:



Product Functions

The Funded Project Tracking system provides online real-time information about the projects and resources for components available with the department and serves as a repository for project status and outcomes. The Product functions are more or less the same as described in the product perspective. The functions of the system include the system providing different types of services based on the type of users.

Provisions for the student/faculty to record their funding applications and project updates on the web-application and ultimately, link their outcomes on the centralized portal. Upon approval by the scrutiny team, the resource manager at the department can reallocate available resources and approve the budget for the remaining components alone.

Website visitors will be provided an interface to view all ongoing internally and externally funded project activities along with updates and outcomes that can be filtered by calendar period, project supervisor, members, domain and budget, and also generate customized reports.

The system uses the University's Google One membership to authenticate users into the system to secure access to the facility to the users.

User characteristics

The users of the system are students, faculty, and resource manager (who is also a delegated faculty) of the university along with administrators who maintain the system. All the users are assumed to have a basic working knowledge of computers and surfing through web pages.

The administrators of the system are expected to have more knowledge of the internals of the system and can rectify the small problems that may arise due to disk crashes, power failures, and other catastrophes to maintain the system.

The proper user interface, users manual, online help, and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

Constraints

- The information of all the faculty/students must be stored in a database that is accessible by the FPT system. This can be linked to the Google Organization account.
- The FPT system is connected to the university intranet and is running throughout the day.
- The users can access the FPT system from any computer that is connected to SSN's intra-network.
- The users must have their correct usernames and passwords to sign in to their organizational accounts hosted on Google One.

Assumptions and Dependencies

- The users have sufficient knowledge of computers.
- The students/faculty must update details of their funded project applications and status on the web-application
- The department's computer should have Internet connection and Internet server capabilities at least sufficient to connect to the local intra-network
- The project outcomes must be specified by the project stakeholders or the web administrator

Specific Requirements

This section describes in detail all the functional and non-functional requirements of the system.

Functional

Updating Project Details on the Web-Application

The web-application can be used to update and track internal funding requests. In addition, it will serve as a repository to record the status of externally funded projects affiliated to the college.

Internal Projects Management

Prior to approving funds and allocating resources, the internal funding committee can find resources already available with the department, allocate them to the project and grant funds only for components that aren't already available with the department.

Faculty and students applying for internal funding must update details and status of their applications, detailing aspects about the funding authority, domain of work, budget requested, etc.

Approved projects can then be used by the applicants to record regular progress, directly visible to the research cell. The final outcomes of the projects will be added to the project in the end and archived into the database.

Intermediate and final outcomes of the projects must also be updated by the applicants, and will be archived after completion.

Upon project completion, the newly acquired resources are inventorized on the system and submitted to the department.

External Projects Management

Applicants applying for external funding must update details and status of their applications, detailing aspects about the funding authority, domain of work, budget requested, etc.

Intermediate and final outcomes of the projects must also be updated by the applicants, and will be archived after completion.

Resource Management

The system also maintains a repository of project resources already available with the system. This includes the availability status of the resources.

The department can choose to allocate these resources to newly approved applicants, whilst approving funds for other requirements alone. This ensures efficient usage and economic repurposing of the college's resources.

Faculty/Students, upon project completion, will deposit the newly acquired components with the department. These items will be inventoried on the system for reuse in future.

Report Generation

Summary statistics of applied and approved funding projects can be curated through the system based on multiple filtering criteria.

This can prove useful during research showcases and academic year progress presentations. Furthermore, it can be pivotal in auditing research and academic activities.

Login Capabilities

The system shall provide the users (students and faculty) with login capabilities through Google OAuth

Mobile Devices

The Project Tracking System will also be supported on mobile devices such as cell phones.

Alerts

The system can alert the admin in case of any ambiguity or problem with the system

Non-Functional

Usability

- The system shall allow the users to access the system from the Internet using HTML or its derivative technologies. The system uses a web browser as an interface.
- Since all users are familiar with the general usage of browsers, no specific training is required.
- The system is user-friendly and self-explanatory.

Reliability

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Accuracy

The accuracy of the system is limited by the accuracy of the speed at which the students and faculty use the system

Access Reliability

The system shall provide 100% access reliability.

Resource Utilization

The resource inventory is updated based on the active projects and the projects that have been completed.

Supportability

The system designers shall take into consideration the following supportability and technical limitations.

Internet Protocols

The system shall comply with the TCP/IP protocol standards and shall be designed accordingly.

Information Security Requirement

The system shall support the FPT information security requirements and use the same standard as the FPT information security requirements.

Maintenance

The maintenance of the system shall be done as per the maintenance contract.

Standards

The coding standards and naming conventions will be as per the Indian and US standards.

Performance

- **Response Time**

The information page should be able to be downloaded quickly and details must be updated in real time at regular intervals. The information is refreshed every two minutes. The access time for a mobile device should be less than a minute. The system shall respond to the member in not less than two seconds from the time of the request submission. The system shall be allowed to take more time when doing large processing jobs.

- **Throughput**

The number of transactions is directly dependent on the number of users, who may be the administrators, students, and also the faculty who use the portal

- **Capacity**

The system should at least handle 250 users at a time.

- **Resource Utilization**

The resources are modified according to the user requirements and also according to the requests of the users, if possible

Design Constraints

Software Language Used

The languages that shall be used for coding the FPT System are Node.js, MongoDB, HTML/CSS, JavaScript.

Development Tools

Will make use of the available Visual Studio Code and will be version controlled with Github.

Class Libraries

Will make use of the existing libraries available for Express.js, Mongoose, Tailwind CSS and Bootstrap.

Online User Documentation and Help System Requirements

Online help is provided for each of the features available with the Funded Project Tracking System. The nature of these systems is unique to application development as they combine aspects of programming (hyperlinks, etc) with aspects of technical writing (organization, presentation).

The User Manual describes the use of the system to librarians and Employees. It describes the use of the system on mobile systems. The user manual should be available as a hard copy and also as online help.

An installation document will be provided that includes the installation instructions and configuration guidelines, which is important to a full solution offering. Also, a readme file is typically included as a standard component. The Read Me includes a "What's New With This Release" section, and a discussion of compatibility issues with earlier releases. Most users also appreciate documentation defining any known bugs and workarounds in the readme file

Interfaces

User Interfaces

Will make use of the existing Web Browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge or Safari. The user interface of the system shall be designed as shown in the user-interface prototypes.

Hardware Interfaces

The existing Local Area Network (LAN) will be used for collecting data from the users.

Software Interfaces

A firewall will be used with the server to prevent unauthorized access to the system.

Communications Interfaces

The Funded Project Tracking System will be connected to the Intranet services of the university.

Licensing Requirements

The usage is restricted to only Dept. of CSE at SSN College of Engineering, Chennai.

Legal, Copyright, and Other Notices

The FP Tracking system is a trademark of the CSE department at SSN College of Engineering and cannot be used without its consent.

Applicable Standards

The ISO/IEC 6592 guidelines for the documentation of computer-based application systems will be followed.

Supporting Information

The use-case storyboards or the user-interface prototypes are not available.

Design Strategies Adopted [\(refer to this\)](#)

Modules Implemented

Identification of Modules

Conceptual Real-World Entity Category List

Conceptual Class Category	Description
Functional Unit	Non-tangible unit of functional purpose that may not be physically perceived but contributes to the development of an intellectual (and/or) perceptible system
Documents	Piece of printed matter that serves as a record
Records	An account of the data
Physical Tangible Objects	Objects that can be touched or felt
Roles of People	Set of behaviors among the people with respect to the problem statement
Place	A particular position, point, or area in space

Identification of Real-World Entities

- Login Device
- Resource Allocation Record
- Resource
- Software resources
- Hardware resources
- Project
- Internal Project
- Internal Faculty Project
- Internal Student Project
- External Project
- Funding Organization

- User
- Person Name
- Department
- Admin
- Student/Faculty
- Guest User
- Resource manager
- Storage Room
- Storage Room In-Charge
- QR Code Scanner
- QR Code Label
- ID Card

Elimination of Trivial Entities

Noun Phrase	Elimination Reasoning
Person name	Is an attribute
Department	Is an attribute
Storage Room In-Charge	Duplicate of Resource Manager, Role
ID Card	Irrelevant to the scenario
Login Device	Irrelevant to the scenario
Funding Organization	Out of System Scope, Is an attribute
QR Code Scanner	Out of System Scope, Scanned ID used as attribute
QR Code Label	Out of System Scope, Contained ID used as attribute

Final List of Entities — Modeled as Modules

- Resource Allocation Record
- Resource

- Software Resources
- Hardware Resources
- Project
- Internal Project
- Internal Faculty Project
- Internal Student Project
- External Project
- User
- Admin
- Student/Faculty
- Guest User
- Resource manager
- Storage Room

Categorization of Modules

Class Name	Conceptual Class Category
Resource Allocation Record	Records, Documents
Resource	Functional Unit, Physical Tangible Object
Software Resource	Functional Unit
Hardware Resource	Physical Tangible Object
Project	Documents, Physical Tangible Object
Internal Project	Documents, Physical Tangible Object
Internal Student Project	Documents, Physical Tangible Object
External Student Project	Documents, Physical Tangible Object
External Project	Documents, Physical Tangible Object
User	Roles of People

Admin	Roles of People
Student	Roles of People
Faculty	Roles of People
Guest User	Roles of People
Resource Manager	Roles of People
Storage Room	Place

Identification of Associations between Modules

Association Category list

An **association** is a relationship between (instances of) classes that indicates some meaningful and interesting connection. These are used to show relationships that need to be remembered and preserved for some duration

Association Category	Description
B is a type of A	B inherits the properties of A and adds further functionality
A controls B	A controls and manages the operation of B and its contents
B describes A	B provides a detailed structured description of A
A is given to B	A is allocated to B for use and is expected to be released back
A belongs to B	B claims ownership over A

Associations Identified and Categorization

Source (A)	End (B)	Relationship	Category
Resource Manager	Faculty	Generalization	B is a type of A
User	Resource Manager	Generalization	B is a type of A
User	Admin	Generalization	B is a type of A
User	Guest User	Generalization	B is a type of A

User	Student	Generalization	B is a type of A
User	Faculty	Generalization	B is a type of A
Resource Manager	Storage Room	Aggregation	A controls B
Storage Room	Resource	Aggregation	A contains B
Resource	Software Resource	Generalization	B is a type of A
Resource	Hardware Resource	Generalization	B is a type of A
Software Resource	Software Resource Details	Composition	B describes A
Hardware Resource	Hardware Resource Details	Composition	B describes A
Resource Allocation Record	Resource	Aggregation	A describes B
Resource Allocation Record	Resource Manager	Aggregation	A describes B
Resource Allocation Record	Internal Project	Aggregation	A is given to B
Project Record	Internal Project	Generalization	B is a type of A
Project Record	External Project	Generalization	B is a type of A
Internal Project	Internal Student Project	Generalization	B is a type of A
Internal Project	Internal Faculty Project	Generalization	B is a type of A
Project Record	Student	Aggregation	A belongs to B
Project Record	Faculty	Aggregation	A belongs to B

Definition of Associations and their notations

Generalization

- Relationship that implements the concept of object orientation called inheritance. The generalization relationship occurs between two entities or objects, such that one entity is the parent, and the other one is the child.
- In UML modeling, a generalization relationship is a relationship that implements the concept of object orientation called inheritance
- The generalization relationship occurs between two entities or objects, such that one entity is the parent, and the other one is the child
- The child inherits the functionality of its parent and can access as well as update it
- Generalization relationship is utilized in class diagrams to specify that the child inherits actions, characteristics, and relationships from its parent

Aggregation

- Represents an associative relationship where the child can exist independently of the parent.
- An aggregation is a subset of association, which represents has a relationship
- It is more specific than association
- It defines a part-whole or part-of relationship
- In this kind of relationship, the child class can exist independently of its parent class

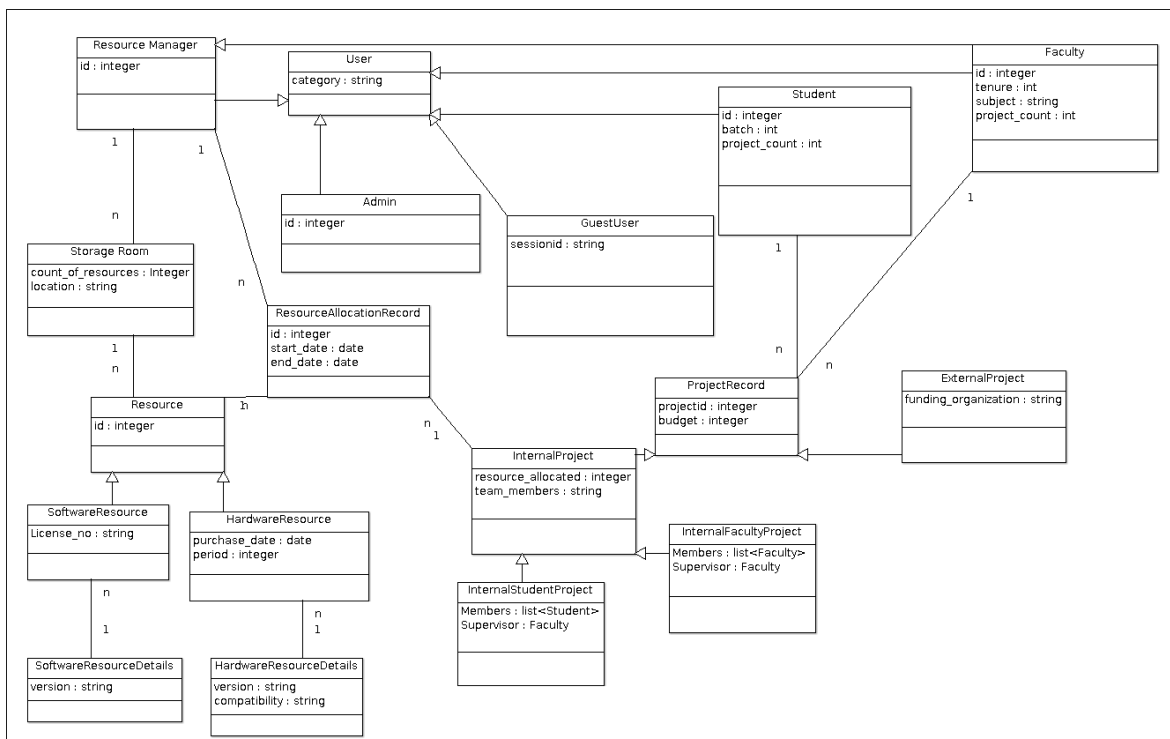
Composition

- Represents an associative relationship where one object is the owner of another object and they are dependent on one another.
- The composition is a subset of aggregation
- It portrays the dependency between the parent and its child, which means if one part is
- deleted, then the other part also gets discarded
- It represents a whole-part relationship

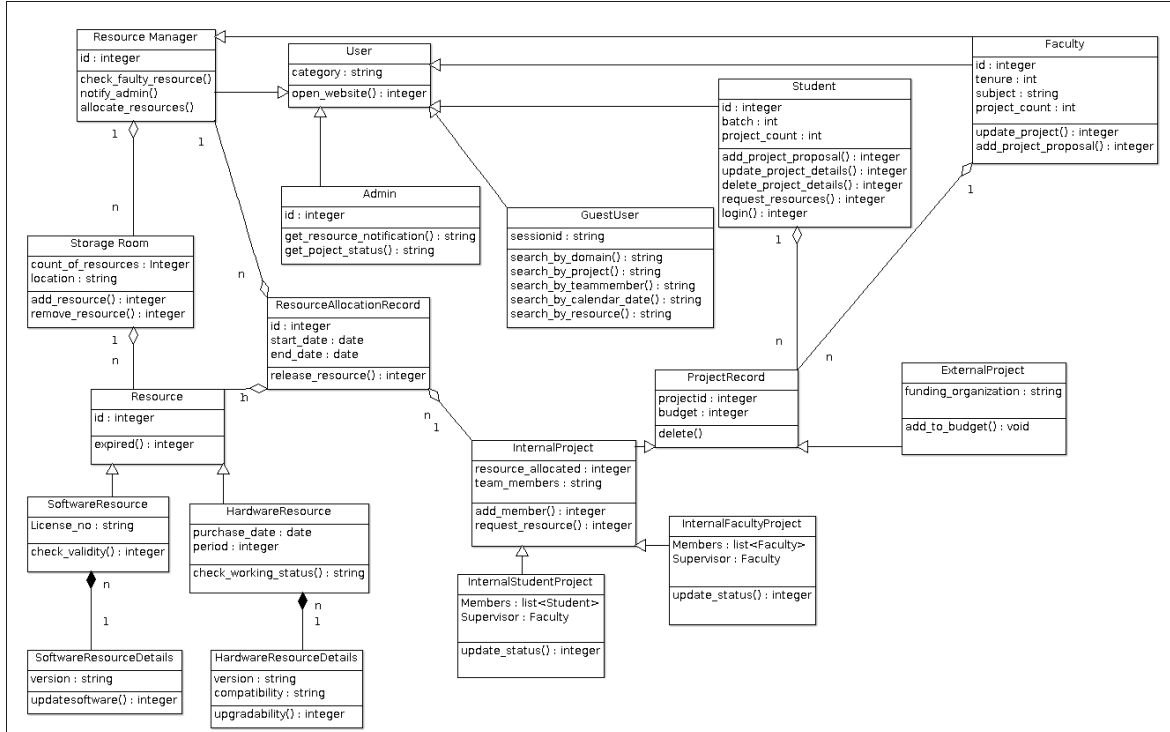
Multiplicity Based Associations

- Multiplicity based associations
- Multiplicity defines how many instances of a class A can be associated with one instance
- of a class B
- It defines a specific range of allowable instances of attributes
- In case if a range is not specified, one is considered as a default multiplicity

Domain View of Modules



Elaborate View of Modules



Logic Representation & Workflow Description

Identified Workflows and System Logic

The following usage scenarios and consequent workflows of the system are identified for the Funded Project Tracking and Resource Management System.

Main Workflow 1: Student/Faculty Registers Project

- Student or faculty member logs into the system
- System validates the log-in credentials
- They register details of the applied projects on the website
- The admin checks the validity of the submitted proposals
- After successful completion of IFP presentations, and the admin gets the approved projects from the Management, the admin updates the application status for the submitted proposals.
- The student/faculty gets notified of their project status

Main Workflow 2: Request for Resources

- Once the applicant gets notified of their project approval, they file their request for the resources for their project.
- The resource manager logs into the system, and can view all the resource requests.
- The Resource manager accesses the inventory, and allocates for the requested resources.
- The admin and the applicant are notified about the allocations, and the corresponding data is updated in the project record by the system.
- The applicant can get the approved resources after mapping his profile with the resource through barcode and biometric/id mapping and use them in their project.

Alternate Workflow 1: Student/Faculty Updates Project Progress

- The applicants can regularly update their project progress through our system
- The system validates the user by their log-in credentials
- The system automatically updates the corresponding project record in the database
- The System sends reminders if the project is at a stale point, and is not updated for a certain period of time.

Alternate Workflow 2: Student/Faculty Records Project Completion

- The applicants log into the system and system validates them
- They provide their project outcome or project completion status to the system for the given project after regular updates done earlier.
- The system updates them to the corresponding project record automatically, and notifies the Resource manager, Admin.
- This is used for reclaiming the resources and the official documentation of the project's completion status.

Alternate Workflow 3: Resource Manager Reclaim Resources

- Once the resource manager is notified of the project completion or resource completion.
- The resource manager gets back the allocated resources from the applicants and checks the conditions of the project.
- Then the resource manager accesses the inventory and update the resources through our system
- The system updates the project record automatically in the database
- The admin updates the IFP status as completed
- The project lead will then be notified for further formalities.

Alternate Workflow 4: Resource Manager Reports Faulty Components

- Sometimes, a resource manager finds a faulty component in the inventory
- Then he updates the inventory and the faultiness is reported to the admin

- The admin gets notified about the faultiness of the corresponding resources mentioned by the resource manager
- The admin can further proceed to place orders or bring in domain experts to handle the problem.

Alternate Workflow 5: Guest Generates/Views Reports

- The guest can view or generate the reports of the projects in our management system
- The reports consist all details entered by the applicants for their corresponding projects
- The dates, subject, and other important information is detailed in the reports
- This is used to generate yearly reports for the magazines , and other official documentation for the department.
- These reports can be generated from the database , as and when required.

Alternate Workflow 6: Guest Searches for Projects

- The guest user can search their desired projects in our system
- They can search based on various filters
 - Search by domain
 - Search by members
 - Search by budget
 - Search by status
 - Search by calendar period
- The guest, or incoming project applicant uses this to extend their ideas and plan on the project ahead of time.
- The admin, uses this to reduce project replications

Identification of Generic System States in the Workflow

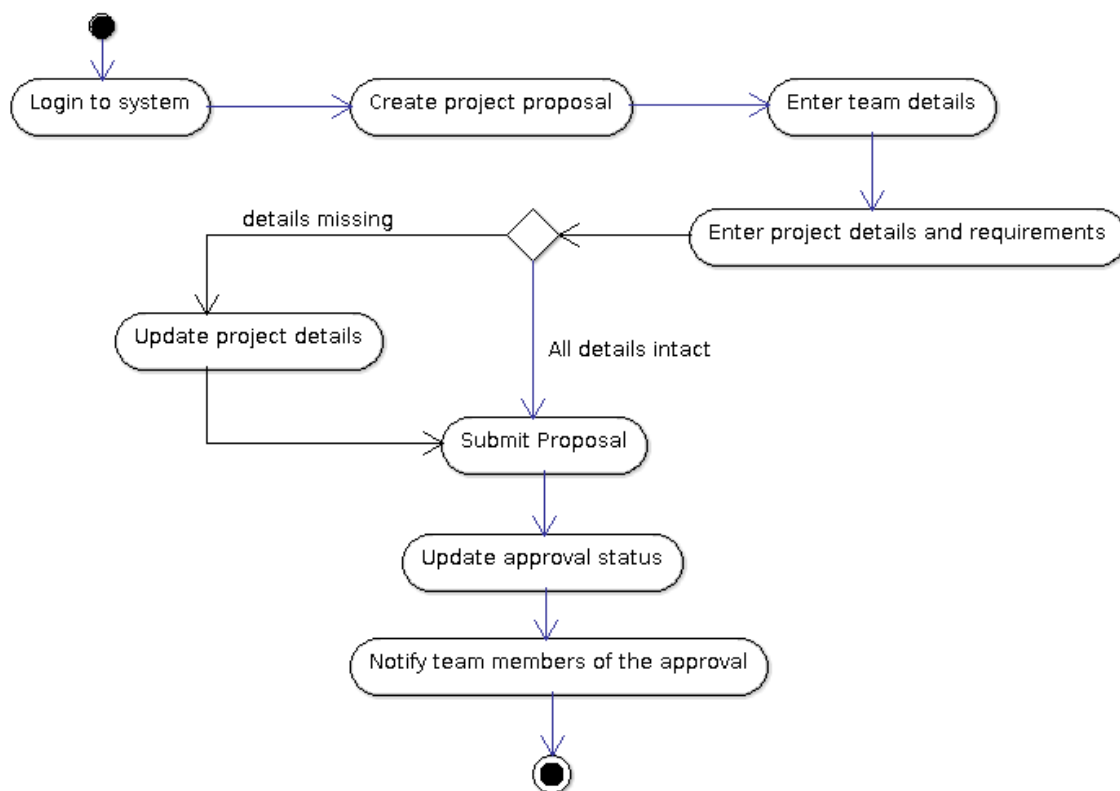
Generic State Template	Container Workflows	Description
Details/Field	1, 2, 3, 5	State that persistently performs data

Entry		verification of entered fields and awaits completion of data entry in real-time
Data Entry Review	1, 2, 3, 5	A review state that facilitates updation/cancellation of the details entered for a specific workflow and allows forwarding or backtracking to its immediate states
Guest Session	1, 2, 3, 4, 5	A dashboard display with response action triggers for a guest user of the website. Also provides the endpoint to transition to a logged-in session
Logged-In Session	1, 2, 3, 4, 5	A dashboard display with response action triggers for a registered user of the website — student, faculty, resource manager or admin
Waiting For User Action	1, 3	State that requires manual intervention to continue to one of the next states on the corresponding workflow
Dispatching Notifications	1, 3	State that prepares and dispatches email and/or dashboard based notification of artifact updates to the concerned stakeholders
Request / Complaint Registration	2, 3	State that compiles entered data and creates an associated entry in the database for further processing
Displaying Summaries	1, 2, 3, 4, 5	State that displays a summary of success or failure for the particular workflow, at the end of all its elemental state transitions and activities
Update / Check Inventory	3, 4, 5	State that searches and interfaces the webpage with the inventory, almost always used only by the resource manager
Archiving Projects	2	State that converts and ongoing project record to a completed project archive, in the database
Retrieving	2, 3	State that interfaces the database to

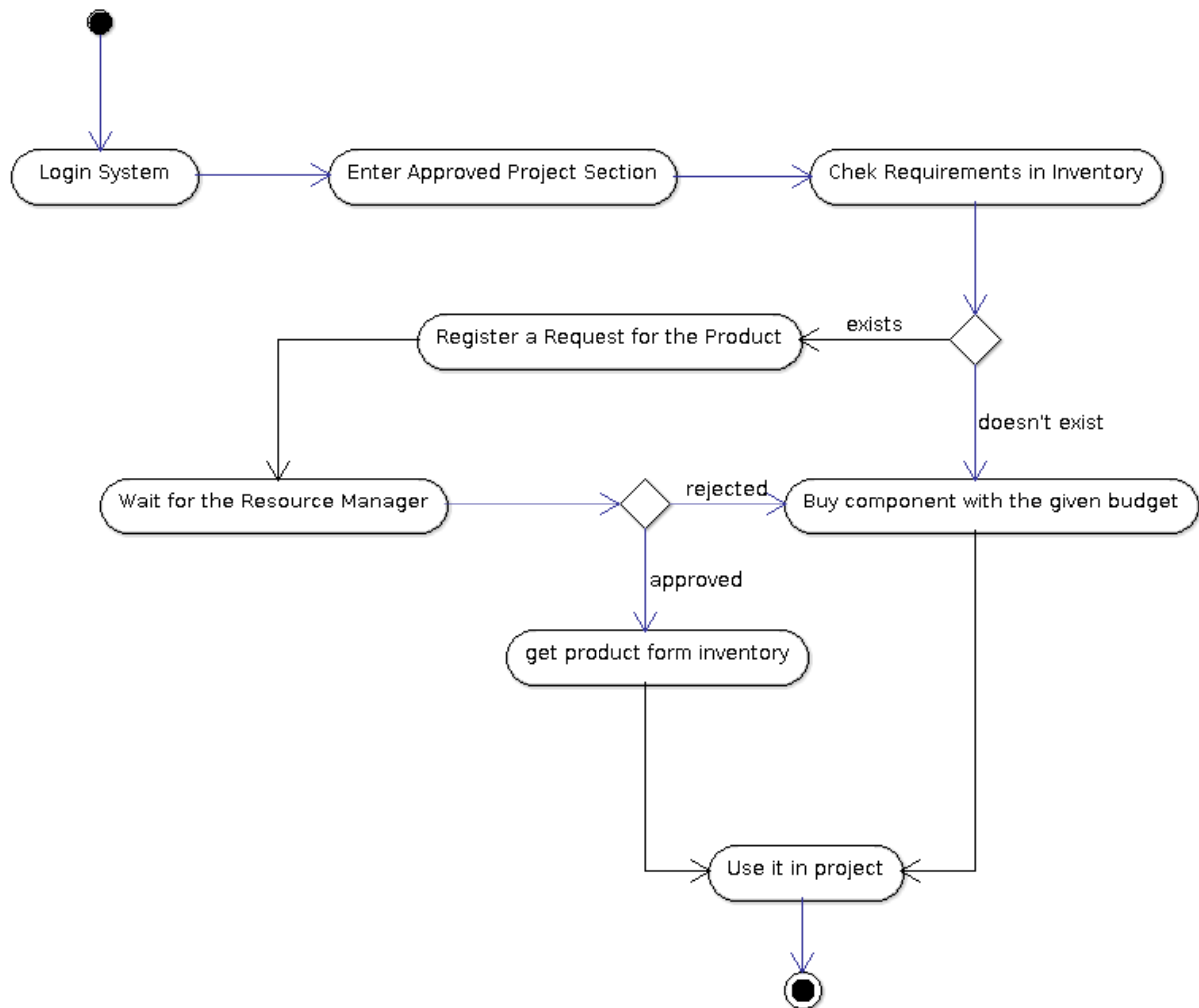
Projects		retrieve projects based on search filter criteria and forwards the data for display or report generation
Generating Reports	5	State that parses details of a display page and prepares a downloadable and printable document format of the data, typically for audit, summary and statistical analysis purposes

Workflows — Flowchart Representation

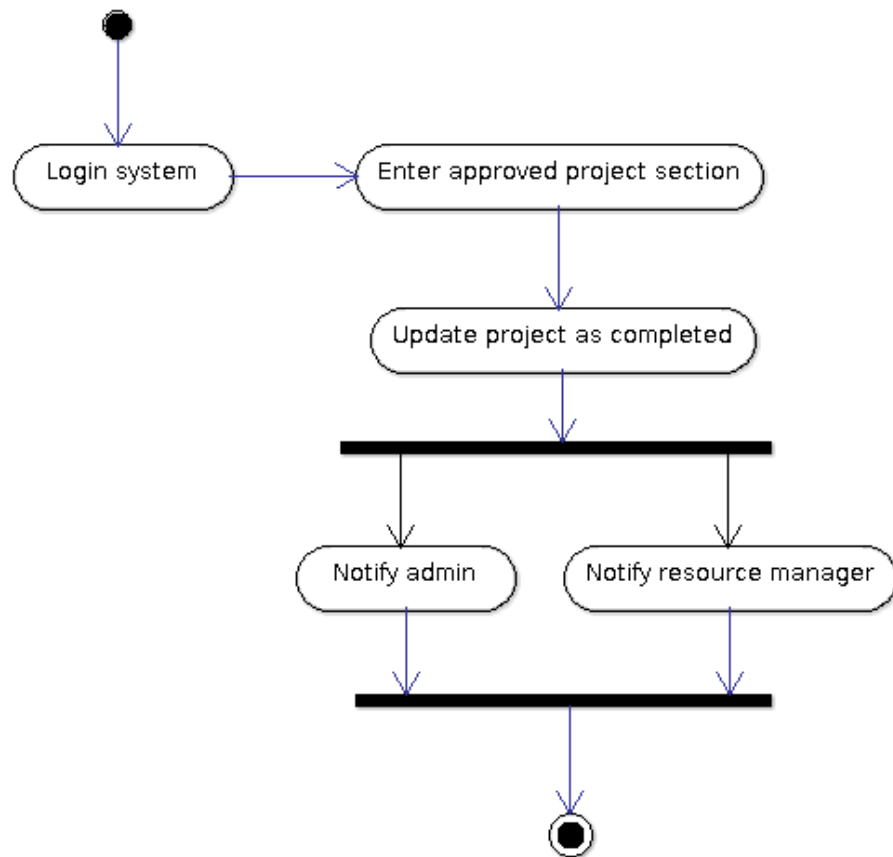
Workflow 1: Record Project Application



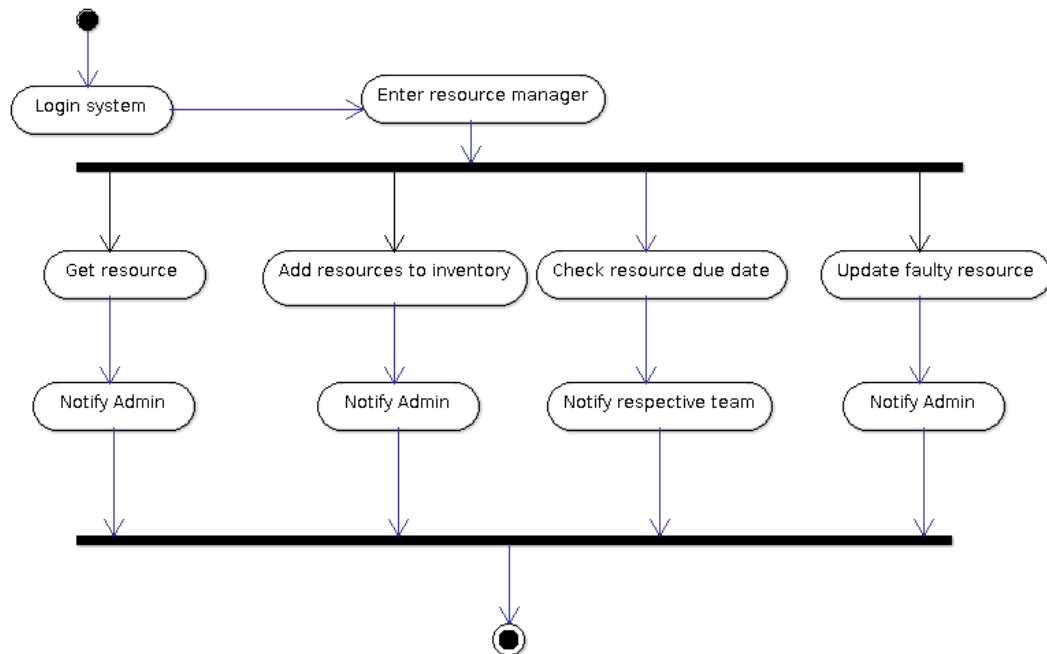
Workflow 2: Resource Allocation Request



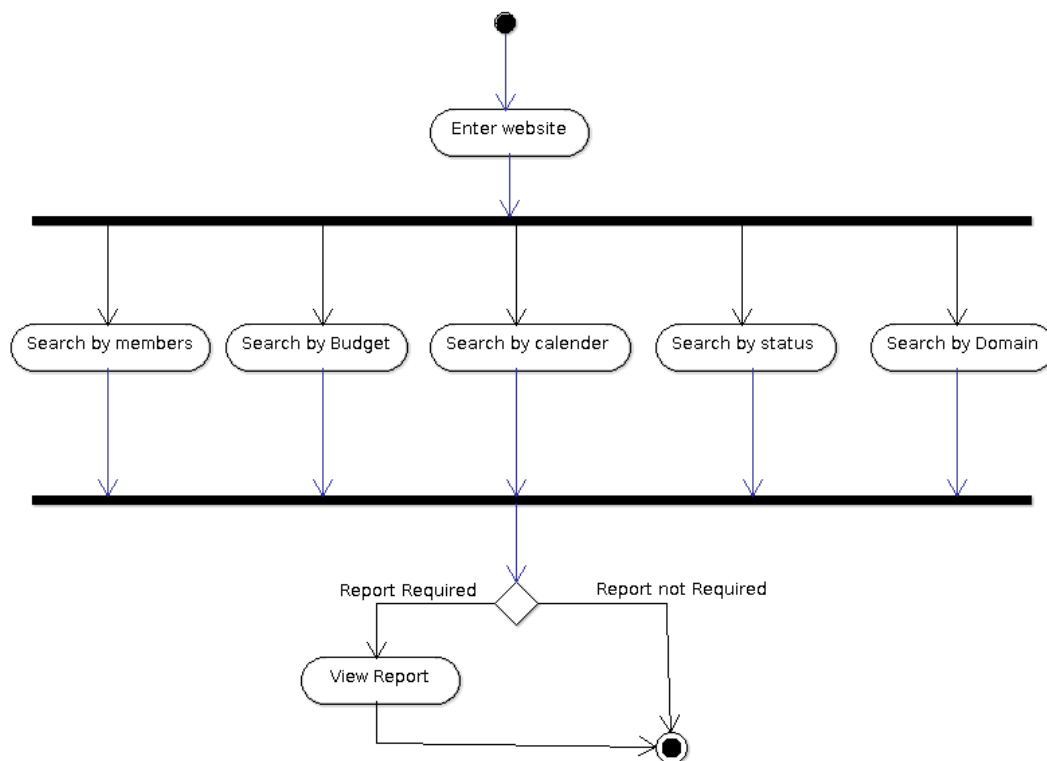
Workflow 3: Project Milestones and Completion Updation



Workflow 4: Resource Management



Workflow 5: Project Search and Report Generation



Components of the Integrated Implementation

Backend — NodeJS with Express

```
├── api-tests
│   ├── project.py
│   ├── proposal.py
│   ├── resource_assignment.py
│   ├── resource_group.py
│   ├── resource.py
│   └── user.py
├── app.js
├── bin
│   └── www
├── check_files
│   ├── agg_doc.pdf
│   └── retrieve_doc.pdf
├── controllers
│   └── api
│       ├── auth.js
│       ├── project.js
│       ├── proposal.js
│       ├── resource_assignment.js
│       ├── resource_group.js
│       ├── resource.js
│       └── user.js
├── db
│   ├── connect.js
│   └── migrations
│       ├── 1652793703064-make-first-admin.js
│       ├── 1652977119177-make-test-users.js
│       └── 1653254944552-make-test-resource.js
├── helpers
│   ├── error.js
│   └── filters
│       ├── resource_group.js
│       └── resource.js
│   ├── resource_group.js
│   ├── resource.js
│   └── utils.js
├── init
│   └── seed_db.js
└── migrate.json
```

models

- index.js
- project.js
- proposal.js
- resource_assignment.js
- resource_group.js
- resource.js
- user.js

package.json

package-lock.json

public

- **images**

- **javascripts**

- **stylesheets**

 - style.css

routes

- auth.js
- index.js
- project.js
- proposal.js
- resource_assignment.js
- resource_group.js
- resource.js
- user.js

unit-tests

- ProposalTests.py
- **__pycache__**
 - ProposalTests.cpython-39.pyc
- run.sh

views

- error.pug
- index.pug
- layout.pug


Frontend — ReactJS with Next

```
.  
├── assets  
│   ├── gifs  
│   │   └── loading.gif  
│   └── imgs  
│       └── user-profile.png  
├── components  
│   ├── FullProjectCard.js  
│   ├── FullProposalCard.js  
│   ├── FullUserCard.js  
│   ├── ProjectCard.js  
│   ├── ProposalCard.js  
│   ├── ResourceAllocationCard.js  
│   ├── ResourceCard.js  
│   ├── styles  
│   │   ├── FullProjectCard.module.css  
│   │   ├── ProjectCard.module.css  
│   │   ├── ProposalCard.module.css  
│   │   ├── ResourceCard.module.css  
│   │   └── UserCard.module.css  
│   └── UserCard.js  
├── next.config.js  
├── package.json  
├── package-lock.json  
├── pages  
│   ├── admin  
│   │   ├── messages.js  
│   │   └── view_unapproved.js  
│   ├── api  
│   │   └── hello.js  
│   ├── _app.js  
│   ├── guest  
│   │   ├── project_page.js  
│   │   ├── proposal_page.js  
│   │   ├── styles  
│   │   │   ├── view_resources.module.css  
│   │   │   └── view_user.module.css  
│   │   ├── user_page.js  
│   │   └── view_project.js
```

```
├── user_page.js
├── view_project.js
├── view_projects.js
├── view_proposals.js
├── view_resources.js
├── view_users.js
├── header.js
├── index.js
├── nothing.js
├── options.js
├── registered_user
│   ├── add_name.js
│   ├── approved_projects.js
│   ├── create_proposal.js
│   ├── messages.js
│   ├── request_resources.js
│   ├── update_project_status.js
│   └── view_approval_status.js
├── resource_manager
│   ├── approve_resource.js
│   ├── messages.js
│   ├── reclaim_resource.js
│   ├── report_faulty.js
│   └── update_inventory.js
├── userpage.js
├── public
│   ├── logo.png
│   └── vercel.svg
├── README.md
├── src
│   ├── assets
│   │   └── loading.gif
│   ├── components
│   │   └── auth
│   │       ├── google_oauth.js
│   │       └── google_oauth.js.old
├── styles
│   ├── globals.css
│   └── Home.module.css
```

Remote Database — MongoDB

Users

 **FPT-Cluster-J2K2** VERSION 5.0.9

[Overview](#) [Real Time](#) [Metrics](#) [Collections](#) [Search](#) [Profiler](#) [Performance Advisor](#) [Online Archive](#) [Cmd Line T](#)

DATABASES: 1 COLLECTIONS: 9

+ Create Database

Q Search Namespaces

FPT_DB

changelog

migrations

project_updates

projects

proposals

resource_assignments

resource_groups

resources

users

FPT_DB.users

STORAGE SIZE: 36KB TOTAL DOCUMENTS: 22 INDEXES TOTAL SIZE: 72KB


[Find](#) [Indexes](#) [Schema Anti-Patterns](#) [Aggregation](#) [Search Indexes](#)

FILTER { field: 'value' }

QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId("628abd684bc531d7264a0a0b")
first_name: "Admin"
last_name: "User"
date_of_birth: 1999-09-09T00:00:00.000+00:00
email: "admin@admin.com"
role: "faculty"
access: Array
deleted_at: null
createdAt: 2022-05-22T22:47:04.532+00:00
```

Proposals

 **FPT-Cluster-J2K2** VERSION 5.0.9 REGION GCP Mumbai

[Overview](#) [Real Time](#) [Metrics](#) [Collections](#) [Search](#) [Profiler](#) [Performance Advisor](#) [Online Archive](#) [Cmd Line Tools](#)

DATABASES: 1 COLLECTIONS: 9 [VISUALIZE YOUR DATA](#)

+ Create Database

Q Search Namespaces

FPT_DB

changelog

migrations

project_updates

projects

proposals

resource_assignments

resource_groups

resources

resources

FPT_DB.proposals

STORAGE SIZE: 10.67MB TOTAL DOCUMENTS: 23 INDEXES TOTAL SIZE: 36KB


[Find](#) [Indexes](#) [Schema Anti-Patterns](#) [Aggregation](#) [Search Indexes](#)

FILTER { _id: ObjectId("628b5425099b1ba80b543da6") } [OPTIONS](#)

```
_id: ObjectId("628b5425099b1ba80b543da6")
title: "IoT Driven Smart Trains"
decription: "blah blah blah..."
domains: Array
supervisors: Array
leader: ObjectId("628abd6a4bc531d7264a0a0")
members: Array
funding_type: "internal"
funding_agency: "IDF Trust"
pdf_document: Binary('JVBERi0xLjQKJCF5j6IKNSAwIG9iag08PC9MZW5ndGggNiAwIFRmIsgVYIC9GbGF0ZUR1Y29kZT4+Cnn0cmVhbQp4nMvaw48d...', 0)
budget: 10000
approved_on: 2022-06-14T05:51:11.953+00:00
rejected_on: null
deleted_at: null
```

9b3b8d370d5c363aeef1#clusters/detail/FPT-Cluster-J2K2

Projects

 **FPT-Cluster-J2K2**

VERSION 5.0.9

OverviewReal TimeMetricsCollectionsSearchProfilerPerformance AdvisorOnline ArchiveCmd Line Tool

DATABASES: 1 COLLECTIONS: 9

+ Create Database

Search Namespaces

FPT_DB

changelog

migrations

project_updates

projects

proposals

resource_assignments

resource_groups

resources

users

FPT_DB.projects

STORAGE SIZE: 36KB TOTAL DOCUMENTS: 11 INDEXES TOTAL SIZE: 36KB


FindIndexesSchema Anti-PatternsAggregationSearch Indexes

FILTER { field: 'value' }

>

```
._id: ObjectId("62a4e23619d791faaf76e19a")
proposal: ObjectId("62a4e20b19d791faaf76e195")
approved_budget: 12000
approved_duration: 18
completed_on: null
status_updates: Array
outcomes: Array
createdAt: 2022-06-11T18:43:02.687+00:00
updatedAt: 2022-06-13T16:54:45.470+00:00
__v: 0
```

Resource Groups

 **FPT-Cluster-J2K2**

VERSION 5.0.9

OverviewReal TimeMetricsCollectionsSearchProfilerPerformance AdvisorOnline ArchiveCmd Line Tool

DATABASES: 1 COLLECTIONS: 9

+ Create Database

Search Namespaces

FPT_DB

changelog

migrations

project_updates

projects

proposals

resource_assignments

resource_groups

resources

users

FPT_DB.resource_groups

STORAGE SIZE: 36KB TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 36KB

FindIndexesSchema Anti-PatternsAggregationSearch Indexes

FILTER { field: 'value' }

QUERY RESULTS: 1-3 OF 3

```
._id: ObjectId("628abd0b4bc531d7264a0aa8")
name: "Arduino UNO"
description: "Microprocessor board with GPIO pins. Version 3"
kind: "hardware"
is_multi_assignable: false
deleted_at: null
createdAt: 2022-05-22T22:47:07.189+00:00
updatedAt: 2022-05-22T22:47:07.189+00:00
__v: 0
```


Resources

FPT-Cluster-J2K2

VERSION 5.0.9

Overview

Real Time

Metrics

Collections

Search

Profiler

Performance Advisor

Online Archive

Cmd Line T

DATABASES: 1 COLLECTIONS: 9

+ Create Database

Q

Search Namespaces

FPT_DB

changelog

migrations

project_updates

projects

proposals

resource_assignments

resource_groups

resources

users

FPT_DB.resources

STORAGE SIZE: 36KB TOTAL DOCUMENTS: 5 INDEXES TOTAL SIZE: 72KB

FindIndexesSchema Anti-PatternsAggregationSearch Indexes

FILTER

{ field: 'value' }

```
_id: ObjectId("628abd6b4bc531d7264a0aac")
resource_group: ObjectId("628abd6b4bc531d7264a0aac")
remarks: "Purchased through Amazon"
scan_code: "ABCD123456"
expiry: null
faulted_at: null
deleted_at: null
createdAt: 2022-05-22T22:47:07.380+00:00
updatedAt: 2022-05-22T22:47:07.380+00:00
__v: 0
```

Resource Assignments

FPT-Cluster-J2K2

VERSION 5.0.9

Overview

Real Time

Metrics

Collections

Search

Profiler

Performance Advisor

Online Archive

Cmd Line T

DATABASES: 1 COLLECTIONS: 9

+ Create Database

Q

Search Namespaces

FPT_DB

changelog

migrations

project_updates

projects

proposals

resource_assignments

resource_groups

resources

users

FPT_DB.resource_assignments

STORAGE SIZE: 36KB TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 36KB

FindIndexesSchema Anti-PatternsAggregationSearch Indexes

FILTER

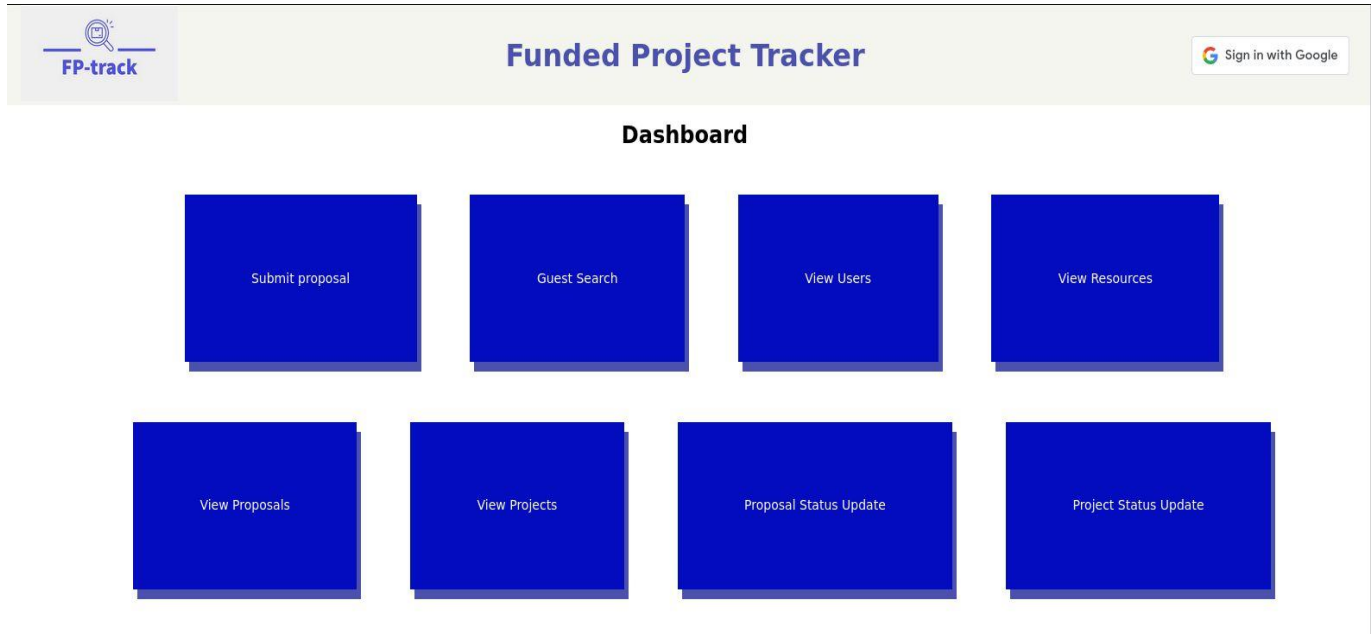
{ field: 'value' }

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId("62a729594264ac72715d90ef")
resource: ObjectId("628abd6b4bc531d7264a0aac")
assigned_to: ObjectId("62a4e23619d791faaf76e19a")
assigned_by: ObjectId("62af1954f5b6a7772b9885b0")
deleted_on: null
createdAt: 2022-06-13T12:11:06.605+00:00
updatedAt: 2022-06-13T12:11:06.605+00:00
__v: 0
```


End-User Interface

User Dashboard




Submit Proposals

The screenshot shows a 'Submit Proposals' form with a teal background. The form fields are arranged vertically: a text input for 'Leader', a text input for 'Members', a dropdown menu for 'Funding Type' currently set to 'Internal', a text input for 'Funding Agency', a file upload section for 'Proposal Document' with a 'Choose File' button and the text 'No file chosen', a text input for 'Budget', and a dark blue 'Submit' button at the bottom.

View Proposals

An overview of pending project proposals with filters by multiple fields




Funded Project Tracker

Sign in with Google

IOT DRIVEN SMART TRAINS internal IDF Trust Rs10000	IOT DRIVEN CLASSROOM internal SSN Trust Rs25000	NUMBER PLATE DETECTION external DST-SERB Rs50000
TEST internal SSN Rs10000	KRTEST1 internal SSN Rs20000	IOT DRIVEN TRAINS internal SSN Trust Rs40000
NUMBER PLATE DETECTION external DST-SERB Rs50000	NUMBER PLATE DETECTION external DST-SERB Rs50000	NUMBER PLATE DETECTION external DST-SERB Rs50000

View Projects - Overview

An overview of ongoing and completed projects with filters by multiple fields



Funded Project Tracker


Sign in with Google

Enter Post Title name

IoT Driven Smart Trains IoT Machine_learning	Number Plate Detection, Again Machine_learning	IoT Driven Smart Trains IoT Machine_learning
IoT Driven Smart Trains IoT Machine_learning	IoT Driven Smart Trains IoT Machine_learning	IoT Driven Smart Trains IoT Machine_learning
IoT Driven Smart Trains IoT Machine_learning	IoT Driven Smart Trains IoT Machine_learning	IoT Driven Smart Trains IoT Machine_learning

View Projects - Detailed

Expanded view of the Principal Investigator of the project



Funded Project Tracker

Sign in with Google

IoT Driven Smart Trains

Domain: iot machine_learning

Budget requested: Rs10000

Budget approved: Rs10000

Duration: 20 months

Proposal.pdf: [link](#)

Outcomes

Status

Principal Investigator

student

user

Yeates, Ben

SSN COLLEGE OF ENGINEERING

ben@gmail.com

Access Rights


user

Members

Supervisors

Resource assigned(collaped)

Expanded view of Members of the project



Funded Project Tracker

Sign in with Google

IoT Driven Smart Trains

Domain: iot machine_learning

Budget requested: Rs10000

Budget approved: Rs10000

Duration: 20 months

Proposal.pdf: [link](#)

Outcomes

Status

Principal Investigator

Members

student

user

Xiao, Claire

SSN COLLEGE OF ENGINEERING

claire@gmail.com

Access Rights

user

student

user

Woakes, Derek

SSN COLLEGE OF ENGINEERING

derek@gmail.com


Access Rights

user

Supervisors

Resource assigned(collaped)

Expanded view of Status Updates of the project



Funded Project Tracker

Sign in with Google

IoT Driven Smart Trains

Domain: iot machine_learning

Budget requested: Rs10000

Budget approved: Rs10000

Duration: 20 months

Proposal.pdf: [link](#)

Outcomes

Status

1. Literature Survey Completed

2. Patent submitted Fully approved design patent made

Principal Investigator


Members

Supervisors

Resource assigned(collaped)

Resource assignable(collaped)

Expanded view of Supervisors of the project



Funded Project Tracker

Sign in with Google

IoT Driven Smart Trains

Domain: iot machine_learning

Budget requested: Rs10000

Budget approved: Rs10000

Duration: 20 months

Proposal.pdf: [link](#)

Outcomes

Status

Principal Investigator

Members

Supervisors

faculty

user

User, Admin

SSN COLLGE OF ENGINEERING

admin@admin.com


Access Rights

admin


Resource assigned(collaped)

Approve/Reject Proposals [Admin]

Admin console to approve or reject proposal based on decision from the funding authority



Funded Project Tracker



funding_agency

internal

KRTEST1

SSN

Rs20000

Accept

Reject

internal

IOT DRIVEN TRAINS

SSN Trust

Rs40000

Accept

Reject

external

NUMBER PLATE DETECTION

DST-SERB

Rs50000

Accept

Reject

external

NUMBER PLATE DETECTION

DST-SERB

Rs50000

Accept

Reject

external

NUMBER PLATE DETECTION

DST-SERB

external

NUMBER PLATE DETECTION

DST-SERB

Interface for resource manager to allocate resources for a project from the inventorized database

Funded Project Tracker

Sign in with Google

IoT Driven Smart Trains

Domain: **iot** **machine_learning**

Budget requested: Rs**10000** Budget approved: Rs**10000** Duration: 20 months Proposal.pdf:[link](#)

- Outcomes
- Status
- Principal Investigator
- Members
- Supervisors

Resource assigned

Filter... name ▾

Resource assignable

Filter... name ▾

Hardware	Software	Hardware
Arduino UNO SSN COLLEGE OF ENGINEERING Microprocessor board with GPIO pins. Version 3 Available quantity: 2 Requesting Quantity: + - Request	Matlab SSN COLLEGE OF ENGINEERING Fully licensed Matlab V1 Available quantity: 3 Requesting Quantity: + - Request	Raspberry Pi 3 SSN COLLEGE OF ENGINEERING Microcontroller board without WIFI built-in Available quantity: 0 Requesting Quantity: + - Request

User Management [Admin]

The screenshot displays the 'FP-track' web application interface. At the top left is the 'FP-track' logo, and at the top right is a 'Sign in with Google' button. The main content area is a grid of user profiles, each with a dark blue background and white text. Each profile includes a role label (e.g., 'faculty', 'student'), a 'user' role indicator, the user's name, affiliation ('SSN COLLEGE OF ENGINEERING'), email address, and an 'Access Rights' section with a role selector dropdown.

Role	User	Email	Access Rights
faculty	User, Admin	admin@admin.com	admin
student	Yeates, Ben	ben@gmail.com	user
student	Xiao, Claire	claire@gmail.com	user
student	Woakes, Derek	derek@gmail.com	user
student	Vincent, Erin	erin@gmail.com	user
faculty	Timothy, Greg	greg@gmail.com	user
faculty	Yeates, Harry	harry@gmail.com	resource_mgr
student	D, Karthik	karthik19047@cse.ssn.edu.in	user, admin, resource_mgr

At the bottom left, a terminal window shows the URL: `localhost:3000/api/user/628abd6a4bc531d7264a0aa2`.

Future Improvements [\(refer to this\)](#)

Features and Techniques not Implemented

Suggested Improvements