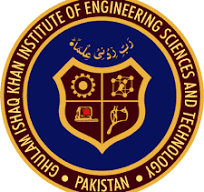
****

**Semester Project Proposal**

**Semester Project Title**:

**FYP Navigator – Revolutionizing Final Year Project Selection**

**Student Details**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student Name | Student Reg | Student Degree |
| Student-1 | Ali Muntazir | 20233098 | Cyber-Security |
| Student-2 | M.Ismail | 2023453 | Cyber-Security |
| Student-3 | Abdullah Ahtasham | 2023033 | Cyber-Security |
| Student-4 | Moeed Niazi | 2023323 | Cyber-Security |

1. **Introduction**

**1.1 Project Overview**

The FYP Navigator is an innovative web-based platform designed to transform the final year project (FYP) selection and collaboration process. The system is conceived as a centralized hub where students, academic supervisors, and coordinators can:

* **Submit and Evaluate Proposals:** Allowing students to propose original project ideas while automatically filtering out duplicate concepts.
* **Facilitate Team Formation:** Helping students form balanced project teams based on shared academic interests and complementary skills.
* **Streamline Supervisor Involvement:** Enabling supervisors and coordinators to review, comment on, and approve project proposals with ease.
* **Enhance Communication:** Providing integrated messaging and notification systems to promote continuous dialogue and updates among all stakeholders.

By addressing common challenges such as manual coordination, project duplication, and inefficient communication, the FYP Navigator aims to foster a more dynamic and innovative academic environment.

* 1. **Objectives**
* **Centralization:** Create a single, unified platform for managing the lifecycle of final year projects—from ideation to approval.
* **Enhanced Efficiency:** Automate key tasks like registration, submission, and status tracking to minimize delays and administrative overhead.
* **Improved Collaboration:** Support seamless interactions among students, supervisors, and coordinators through integrated communication tools.
* **Robust Data Management:** Leverage PostgreSQL for a secure, scalable, and well-structured relational database that maintains data integrity.
* **User-Centric Design:** Develop an intuitive, responsive interface that caters to different user roles (students, supervisors, coordinators) and device types.
* **Innovation Enforcement:** Incorporate smart algorithms to detect and prevent duplicate submissions, ensuring the originality of every project idea.

1. **Roadmap (Project Lifecycle)**

**2.1 Phase 1: Planning & Requirements Gathering**

* **Scope Definition:** Clearly outline the boundaries of the project, detailing the features and functionalities required for FYP selection and management.
* **Stakeholder Analysis:** Identify and engage with all relevant stakeholders (students, faculty, administrative staff) to gather diverse perspectives on needs and expectations.
* **Requirement Collection:** Conduct surveys, interviews, and review existing literature to document both functional and non-functional requirements.
* **Technology Assessment:** Evaluate potential DBMS options, web frameworks (such as Flask for backend operations), and front-end technologies.
* **Project Scheduling:** Define milestones, deadlines, and resource allocations; set up JIRA to track tasks and progress.
  1. **Phase 2: Database Design**
* **ER Modeling:** Develop an in-depth Entity-Relationship Diagram that captures:
  + **Users:** Central table containing user credentials and roles.
  + **Students & Faculty:** Extend the user model with additional attributes such as skill sets and job titles.
  + **Project Description:** A detailed repository for project proposals, including title, description, and associated supervisor.
  + **Projects and Group Members:** Separate tables to manage project metadata and team compositions.
* **Schema Design & Normalization:** Define primary keys, foreign keys, and relationships ensuring minimal redundancy and efficient query performance.
* **Data Integrity Measures:** Implement constraints and triggers to maintain consistent, valid data across all tables.

**2.3 Phase 3: System Design & Development**

* **Architecture Design:** Outline the overall system architecture:
  + **Backend:** Built using Flask (Python) to handle routing, business logic, and database integration.
  + **Database:** PostgreSQL serving as the robust backbone for data management.
  + **Frontend:** Develop using HTML/CSS/JavaScript, optionally enhanced with responsive frameworks like Bootstrap or Tailwind CSS.
* **Module Breakdown:**
  + **User Management Module:** Encompasses registration, login, and session control with role-based access.
  + **Project Registration & Approval Module:** Allows students to submit project proposals and coordinators to review and update statuses.
  + **Dashboard Module:** Customized dashboards for different roles to monitor ongoing projects, approvals, and team status.
* **UI/UX Design:** Create wireframes and prototypes that emphasize ease-of-use, clarity, and accessibility. Iteratively refine based on stakeholder feedback.
* **Integration Strategy:** Develop APIs to ensure seamless communication between frontend, backend, and the PostgreSQL database.
  1. **Phase 4: Testing & Debugging**
* **Unit Testing:** Write comprehensive tests for each module (e.g., login, project submission) to verify individual functionalities.
* **Integration Testing:** Validate the interactions between modules—ensuring that the frontend correctly consumes backend services and that the database queries return expected results.
* **Security Testing:** Perform thorough penetration testing to ensure that authentication, authorization, and data encryption measures are effective against threats.
* **Load & Performance Testing:** Simulate high-usage scenarios to assess the system’s scalability and responsiveness, identifying bottlenecks for future optimization.
  1. **Phase 5: Deployment & Maintenance**
* **Deployment Strategy:** Plan for a secure rollout on cloud services (such as AWS or Firebase) with auto-scaling capabilities.
* **User Onboarding & Training:** Develop detailed documentation, video tutorials, and interactive guides to help users navigate the system.
* **Maintenance Plan:** Schedule regular backups, security updates, and performance monitoring. Gather user feedback post-deployment to guide iterative improvements.
* **Future Enhancements:** Identify potential upgrades like integrating AI-driven project recommendations, real-time chatbots, and further UI enhancements to continuously improve the user experience.

1. **Functional Requirements**

**3.1 User Management**

* **Registration:**
  + Input: User ID, full name, registration number, chosen PIN, role (student, supervisor, coordinator), faculty, and additional details such as skill set for students.
  + Process: Validate and store user data in a secure database.
  + Output: Confirm registration and provide login credentials.
* **Authentication:**
  + Input: Username and PIN.
  + Process: Validate credentials and initiate a session if successful.
  + Output: Redirect to a role-specific dashboard.
* **Session Management:**
  + Maintain persistent sessions and provide secure logout functionality.
  1. **Project Registration & Management**
* **Project Proposal Submission:**
  + Input: Project title, description, required skills, supervisor details.
  + Process: Store the proposal in the project description table and flag it for review.
  + Output: Confirmation message with a unique project ID.
* **Project Approval Process:**
  + Role: Coordinators review pending proposals.
  + Input: Project ID and decision (approve/reject).
  + Process: Update project status in the database and notify relevant parties.
  + Output: Display updated project lists and confirmation of changes.
* **Team Formation:**
  + Enable students to form groups and associate multiple user IDs with a single project.
  + Display team compositions on individual dashboards.

**3.3 Project Viewing & Communication**

* **Project Dashboard:**
  + Provide separate views for pending, approved, and personal project submissions.
  + Allow filtering and sorting by status, supervisor, or submission date.
* **Notification System:**
  + Implement in-app notifications and optional email alerts for status updates and deadlines.
* **Messaging Module:**
  + Integrate real-time chat functionalities to facilitate discussions between team members and supervisors.

1. **Non-Functional Requirements**

**4.1 Performance & Scalability**

* **Database Optimization:**
  + Use indexing, query optimization, and proper normalization to handle increasing data volumes.
* **Responsive Design:**
  + Ensure the web interface adapts seamlessly to desktops, tablets, and smartphones.
* **Cloud Scalability:**
  + Leverage cloud hosting (AWS/Firebase) to handle load spikes and provide reliable uptime.

**4.2 Security & Data Privacy**

* **Encryption:**
  + Encrypt sensitive user data both at rest and during transmission.
* **Access Control:**
  + Implement strict role-based access control (RBAC) to limit functionality based on user roles.
* **Compliance:**
  + Adhere to data protection regulations (e.g., GDPR) and conduct regular security audits.

**4.3 Usability & Accessibility**

* **User-Centric Design:**
  + Design the UI with clear navigation, minimal clutter, and intuitive interactions.
* **Accessibility Standards:**
  + Ensure compatibility with assistive technologies and compliance with WCAG guidelines.
* **Error Handling:**
  + Provide meaningful error messages and support resources to guide users in resolving issues.

**4.4 Reliability & Maintainability**

* **Backup and Recovery:**
  + Schedule automated backups and define clear recovery protocols in case of data loss.
* **Modular Architecture:**
  + Structure the application in modular components to facilitate easier updates, bug fixes, and scalability enhancements.
* **Documentation:**
  + Maintain comprehensive technical and user documentation to support ongoing maintenance and future development.

1. **Project Management (Using JIRA)**

**5.1 JIRA Setup & Workflow**

* **Project Initialization:**
  + Define high-level epics covering database design, frontend and backend development, testing, and deployment.
* **User Stories and Task Breakdown:**
  + Break down each epic into detailed user stories, ensuring that tasks are clearly defined with acceptance criteria.
* **Task Assignment & Scheduling:**
  + Assign tasks to team members, establish deadlines, and set up sprints for iterative development.
* **Progress Tracking:**
  + Use JIRA boards (To-Do → In Progress → Code Review → Testing → Deployment) to visualize task progression.
* **Daily Stand-ups & Reviews:**
  + Hold daily scrum meetings to address blockers, review progress, and adjust plans as needed.
* **Issue Resolution:**
  + Track bugs and enhancement requests in JIRA, documenting fixes and improvements in subsequent sprints.

1. **Tools & Frameworks**

**6.1 Database & Backend**

* **Database Management System:** PostgreSQL
  + Ensures data consistency, robustness, and supports complex queries required for project management.
* **Backend Framework:** Flask (Python)
  + Provides a lightweight yet powerful framework for handling routing, authentication, and API integration.

**6.2 Frontend & User Interface**

* **Frontend Technologies:**
  + HTML, CSS, and JavaScript to build dynamic, responsive interfaces.
  + Optional frameworks such as Bootstrap or Tailwind CSS for rapid UI development and consistency.
* **UI/UX Tools:**
  + Use design tools like Figma or Adobe XD to prototype and iterate on the user interface design.

**7. Conclusion**

The FYP Navigator is positioned to transform the way final year projects are managed and executed by establishing a centralized, user-friendly, and robust platform. This detailed proposal outlines a comprehensive roadmap—from planning and requirements gathering through design, development, testing, and deployment—to ensure that every aspect of the project is addressed systematically. With a focus on innovative features, data security, and continuous improvement, the FYP Navigator is not only an effective tool for today’s academic challenges but also a scalable foundation for future enhancements. This project will ultimately empower students, streamline supervisor involvement, and foster a collaborative environment that drives academic excellence and innovation.