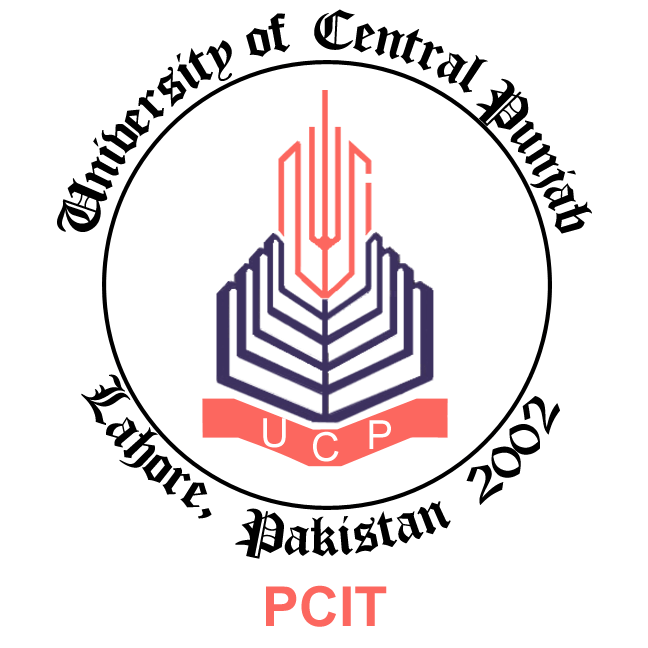
**BSCS FINAL PROJECT PROPOSAL**

**Revolutionizing Performance Management: Perfomix System**

*Term of Registration: Fall 2024*



Presented by:

|  |  |
| --- | --- |
| **Registration No:** | **Name:** |
| L1F21BSCS0005 | EZAAN AMIN |
| L1F21BSCS0937 | Mahmood Amir |
| L1F21BSCS0015 | Farhan Tayyab Ch. |

University of Central Punjab

|  |
| --- |
| Faculty of Information Technology |

**Project Title**

**Revolutionizing Performance Management: Perfomix System**

**Project Advisor**

Muhammad Noman

**Particulars of the students:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Registration#**  eg.**L1F00BSCS0101** | **Name in Full**  Use Block Letters | **CGPA** | **Signatures** |
| 1 | L1F21BSCS0005 | EZAAN AMIN | 2.05 |  |
| 2 | L1F21BSCS0937 | Mahmood Amir | 3.32 |  |
| 3 | L1F21BSCS0015 | Farhan Tayyab Ch. | 2.49 |  |

**Advisor’s Consent**

I Prof./Dr./Mr./Ms.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Muhammad Noman\_\_\_\_\_\_\_\_\_** am willing to guide these students in all phases of above-mentioned project as advisor. I have carefully seen the Title and description of the project and believe that it is of an appropriate difficulty level for the number of students named above.

|  |  |  |
| --- | --- | --- |
| **Note:**  Advisor can’t be changed without prior permission of the Manager Projects and the duration for completion of the Project is 2 regular semesters (approx.) from the date of Registration of Research Project. | Signatures and Date  |  | | --- | |  |   **Advisor** |

**EVALUATOR/REFEREE 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I have carefully read the project proposal and feel that the proposed project is a useful one and of a sufficient difficulty level to justify 2 regular semesters workload for above mentioned students. I have made recommendations in the evaluation form to improve the scope and quality of the project. | | | | | |
|  | | | | Signatures and Date | |
|  |  |  |  |  |  |
|  | | | |  |

**EVALUATOR/REFEREE 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I have carefully read the project proposal and feel that the proposed project is a useful one and of a sufficient difficulty level to justify 2 regular semesters workload for above mentioned students. I have made recommendations in the evaluation form to improve the scope and quality of the project. | | | | | |
|  | | | | Signatures and Date | |
|  |  |  |  |  |  |
|  | | | |  |

**Abstract / Executive Summary**

The **Perfomix System** is an innovative automated performance management solution tailored to streamline and enhance employee evaluation processes across various departments. The system allows users to create customizable performance matrices, define KPIs, and generate comprehensive reports. Perfomix integrates modern tools like GitHub, Jira, Google Sheets, and SQL for seamless data flow, automated performance tracking, and efficient evaluations. Moreover, the system features an NLP-based Recommendation System, providing personalized improvement suggestions for employees based on performance data.

The problem Perfomix addresses is the lack of flexibility in existing performance management systems, which often fail to adapt to an organization's unique needs. By automating key performance indicators (KPIs) tracking and utilizing anomaly detection, Perfomix significantly reduces manual intervention and ensures real-time performance monitoring. The end result is a data-driven, efficient, and user-friendly platform that enhances decision-making and fosters employee growth.

### Introduction and Background

The **Perfomix System** is designed to address challenges in performance management by providing a flexible, automated solution that adapts to various organizational needs. Many traditional performance evaluation tools lack the customization required to meet specific departmental requirements, leading to inefficient feedback mechanisms and limited employee development. Perfomix seeks to solve these issues by integrating widely-used software tools (e.g., GitHub, Jira, Google Sheets) and automating KPI tracking, enabling a seamless evaluation process.

### Statement of the Problem

Traditional performance management systems are often rigid and unable to adjust to the specific requirements of different departments within an organization. As a result, they offer ineffective feedback and fail to promote meaningful employee development. This lack of adaptability leads to inefficiencies, limiting organizational growth and employee performance. Perfomix addresses this problem by offering a customizable and automated performance management system, ensuring seamless evaluations and more personalized employee improvement suggestions through data-driven insights.

### Objectives / Aims / Targets

1. **Automate KPI Tracking:** Create a system that automatically tracks performance metrics for software engineers, project managers, business managers, and testing teams using API integrations.
2. **Provide Real-Time Anomaly Detection:** Detect performance anomalies in real-time and notify relevant stakeholders.
3. **NLP-Based Recommendations:** Use feedback and performance data to provide actionable recommendations for employee improvement.
4. **Seamless User Experience:** Ensure that Admins, Line Managers, and Staff can easily navigate and use the platform for performance evaluations.
5. **Generate Detailed Reports:** Enable users to produce comprehensive performance reports that aid decision-making processes.

**Completeness Criteria**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Criteria** | **Weightage %** |
| 1 | Automated KPI tracking for various roles (Software Engineers, Project Managers, Business Managers, Testing teams) | 30% |
| 2 | Real-time anomaly detection for performance metrics | 20% |
| 3 | NLP-based personalized recommendation system | 25% |
| 4 | User-friendly UI for Admin, Line Manager, and Staff roles | 10% |
| 5 | Detailed performance reports and analytics | 15% |

### Challenges

1. **Complex API Integration:** Integrating APIs from various tools (e.g., GitHub, Jira, Google Sheets) for automated data collection.
2. **NLP for Recommendations:** Developing an effective NLP-based recommendation system that provides actionable employee improvement suggestions.
3. **Anomaly Detection:** Implementing an anomaly detection system that accurately identifies performance dips across multiple departments.
4. **User Interface Complexity:** Designing an intuitive UI that caters to different user roles (Admin, Line Manager, Staff).

**Knowledge Areas Required**

**Software Engineering:** For system architecture and backend development using Flask.

* **Machine Learning / NLP:** For implementing the NLP-based recommendation system.
* **Database Management:** For designing and managing MySQL databases.
* **Data Integration:** Utilizing APIs to fetch performance data from tools like GitHub and Jira.
* **Performance Management:** Understanding of KPIs and performance evaluation metrics.

**Learning Outcomes**

**Expertise in React JS for UI development.**

* Deep understanding of **Flask** for backend services.
* Experience with **MySQL** databases for performance metrics storage.
* Proficiency in **NLP-based systems** for providing actionable employee feedback.
* Knowledge of integrating **APIs** from tools like GitHub, Jira, and Google Sheets.

**Nature of the End Product / Research Outcomes**

The end product will be a **customizable performance management system** that integrates APIs from various tools, automates performance evaluations, and provides data-driven recommendations for employee improvement. It will feature anomaly detection, KPI tracking, and an intuitive user interface catering to multiple roles (Admin, Line Manager, Staff).

**Related Work / Literature Survey / Literature Review**

Performance management systems in use today largely lack the necessary adaptability for various departmental needs. Some existing systems automate KPI tracking but do not provide data-driven feedback or personalized improvement suggestions. Perfomix stands out by combining anomaly detection and NLP-based recommendations, offering a solution that adapts to specific organizational needs and promotes continuous employee growth.

**Deliverables / Work Breakdown Structure**

**API Integrations:** Automate data collection from GitHub, Jira, and Google Sheets for performance tracking.

* **Anomaly Detection Module:** Implement a real-time anomaly detection system.
* **NLP Recommendation System:** Develop an NLP-based system to provide actionable employee feedback.
* **UI/UX Development:** Build a user-friendly interface for various roles.
* **Database Management:** Set up MySQL databases to store and retrieve performance metrics.

**Project Plan / Project Schedule / Project Timetable / Project Calendar**

**Week 1-2: Project Setup and Initial Architecture**

* Set up the project repository on GitHub.
* Design and finalize project architecture.
* Set up backend services using Flask.
* Initialize MySQL database schema and structure for storing performance metrics.
* Ensure basic functionality of the database with test data.

**Week 3-4: API Integrations (Part 1)**

* Integrate GitHub and Jira APIs for automated KPI tracking of Software Engineers and Project Managers.
* Implement basic data collection from these APIs.
* Test the API integrations with test data to ensure proper collection of KPIs.

**Week 5-6: API Integrations (Part 2)**

* Integrate Google Sheets API for Business Managers and CRM tools for client-related metrics.
* Finalize all API integrations.
* Implement secure authentication and authorization mechanisms for data access.
* Refine API calls to ensure data accuracy and completeness.

**Week 7-8: Implement Real-Time Anomaly Detection**

* Develop and implement algorithms to detect performance anomalies using the collected KPI data.
* Test anomaly detection with test scenarios for various departments.
* Build an alert system to notify admins or line managers when anomalies occur.

**Week 9-10: Develop NLP-based Recommendation System**

* Implement the Natural Language Processing (NLP) model for analyzing performance feedback.
* Create a recommendation engine that suggests actionable improvements based on feedback data.
* Fine-tune the NLP model to ensure accuracy and relevance of the recommendations.
* Test the recommendation system with sample data.

**Week 11: UI/UX Development (Part 1)**

* Begin front-end development using React JS.
* Design and develop basic UI components for Admin, Line Manager, and Staff roles.
* Ensure responsive design for various device types.

**Week 12-13: UI/UX Development (Part 2)**

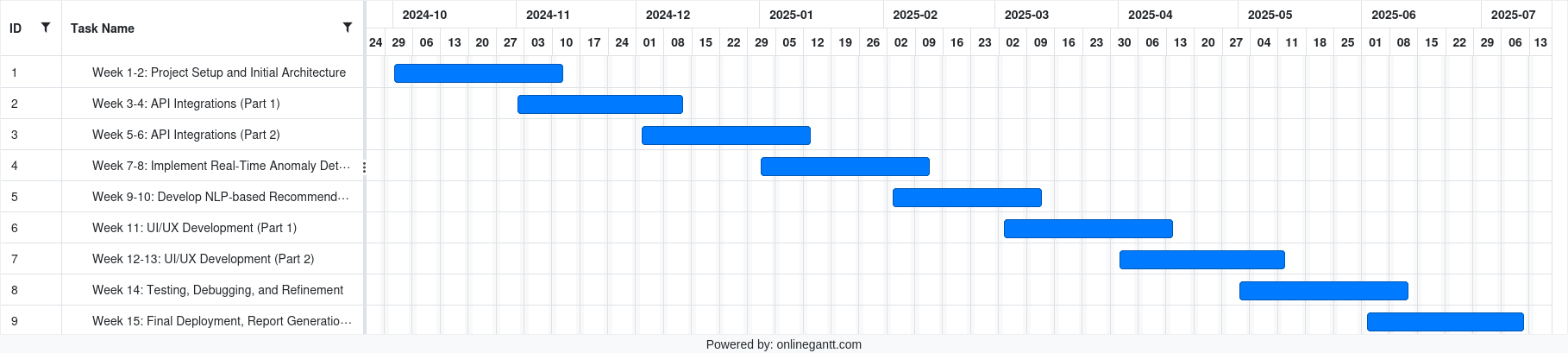
* Complete the UI/UX development for all roles and features.
* Integrate front-end components with the Flask backend and APIs.
* Ensure the UI displays real-time performance data and anomaly alerts.

**Week 14: Testing, Debugging, and Refinement**

* Perform thorough testing of all modules (API integrations, anomaly detection, NLP recommendations, and UI).
* Identify and fix bugs or issues in both the front-end and back-end components.
* Conduct performance testing to ensure scalability and responsiveness of the system.

**Week 15: Final Deployment, Report Generation, and Presentation**

* Deploy the Perfomix system on a cloud server (AWS, Heroku, etc.).
* Generate sample performance reports for Software Engineers, Project Managers, and Business Managers.
* Prepare final documentation including system architecture, test cases, and user manual.
* Conduct a final project presentation and demo of Perfomix.



**Resources Required**

**Development Tools:** Flask, React JS, MySQL, GitHub, Jira, Google Sheets.

* **Hardware:** Development servers and testing environments.
* **Software Libraries:** NLP libraries (e.g., spaCy, NLTK), anomaly detection libraries.

### ****Miscellaneous****

* **Project Management Tools:** Utilize tools like Trello or Jira for task management and progress tracking. Regular stand-up meetings will help keep the team aligned and address any roadblocks.
* **Version Control:** Use Git for version control to ensure collaborative development. Establish branching strategies to manage features and releases effectively.
* **Documentation:** Maintain comprehensive documentation throughout the project for easier onboarding of new team members and future reference. Use tools like Markdown or Confluence for creating and sharing documentation.
* **Testing Strategies:** Incorporate unit testing and integration testing in the development process to ensure all components work correctly. Consider using frameworks like PyTest for backend testing and Jest for front-end components.
* **Feedback Loop:** Implement regular feedback sessions with stakeholders to validate features and adjust the project scope as necessary.

### ****Abstract Story Board and Identification of Characters (For Game-Oriented Projects Only)****

This section is not applicable for the Perfomix project, as it is not game-oriented.

### ****Sketch of Proposed Solution (For Research-based and Hardware-Oriented Projects Only)****

Since this project is software-oriented, you can include a high-level diagram instead:

#### ****High-Level System Architecture Diagram:****

* **Components to include in the diagram:**
  + **Frontend (React JS)**
    - User Interfaces for Admin, Line Manager, Staff
  + **Backend (Flask)**
    - API endpoints for data collection and retrieval
  + **Database (MySQL)**
    - Performance metrics storage
  + **API Integrations**
    - GitHub API
    - Jira API
    - Google Sheets API
  + **Real-Time Processing Layer**
    - Anomaly Detection Module
    - NLP Recommendation Engine

(You would typically create this diagram using a drawing tool like Lucidchart, Draw.io, or even hand-drawn if acceptable.)

### ****References/Bibliography****

1. **Books and Articles:**
   * Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
   * Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.
2. **Online Courses:**
   * Andrew Ng. “Machine Learning.” Coursera, Stanford University.
   * “Deep Learning Specialization.” Coursera, Andrew Ng.
3. **API Documentation:**
   * GitHub API Documentation. Available at: [GitHub API](https://docs.github.com/en/rest)
   * Jira REST API Documentation. Available at: Jira API
   * Google Sheets API Documentation. Available at: Google Sheets API
4. **Tools and Frameworks:**
   * Flask Documentation. Available at: Flask Docs
   * React Documentation. Available at: React Docs
5. **Research Papers:**
   * Zhang, S., et al. “A Review on Anomaly Detection Techniques in Financial Transactions.” IEEE Access, vol. 8, 2020, pp. 134451-134466.
   * Vaswani, A., et al. “Attention is All You Need.” Advances in Neural Information Processing Systems, vol. 30, 2017.