for

Software Requirements Specification

**Version 1.0**

Date created : 20th October 2024

*Department of Computer Science*

*Aligarh Muslim University*

Prepared by : *Mohd Saud*

23CAMSA107

GK0827

2024-2025

Automating Lab Management using GitHub

**LabTracker**

Table of Contents

**Table of Contents……………………………………………………………….………..…………….1**

[1. Introduction 2](#_Toc180268810)

[1.1 Purpose 2](#_Toc180268811)

[1.2 Document Convention 2](#_Toc180268812)

[1.3 Intended Audience and Reading Sugesstions 2](#_Toc180268813)

[1.4 Project Scope 3](#_Toc180268814)

[1.5 References 3](#_Toc180268815)

[2. Overall Description 4](#_Toc180268816)

[2.1 Product Perspective 4](#_Toc180268817)

[2.2 Product Features 4](#_Toc180268818)

# 1. Introduction

## 1.1 Purpose

The purpose of **LabTracker** is to automate the lab management process for the Department of Computer Science, AMU. The system enables students to submit programming problems via GitHub and helps teachers track their progress efficiently. By integrating with the GitHub API, LabTracker provides teachers with insights into students' commitment patterns and allows for the generation of various reports to monitor class performance. It streamlines the workflow for teachers by automating tasks like problem submission tracking and deadline management.

## 1.2 Document Convention

This document uses the following conventions:

- **Bold**: Indicates key terms or concepts that are important within the context of the document.

- ***Italic*:** Used for emphasis or to denote specific titles, names of documents, applications, or sections.

- **List Items**: Bullet points are used to enumerate features, requirements, or important notes for clarity and easy reference.

- **Headings**: Major sections are marked with headings of different levels to structure the document and aid in navigation.

**- References**: URLs and external resources are provided in a consistent format for easy access and further reading.

## 1.3 Intended Audience and Reading Sugesstions

This document is intended for:

* **Project Stakeholders**: This includes faculty members and department heads who will oversee the implementation of LabTracker.
* **Developers**: As the sole developer, this section will guide you in designing, implementing, and testing the application.
* **End Users**: Students and teachers who will interact with the system.

Reading Suggestions:

* Stakeholders should focus on sections detailing the project scope and system overview.
* Developers should pay close attention to functional and non-functional requirements.
* End users might find the user interface descriptions and functionalities most relevant.

## 1.4 Project Scope

**LabTracker** is a web-based application aimed at automating lab management for the Department of Computer Science at AMU. It enables students to submit programming assignments via GitHub, allowing teachers to efficiently track progress and generate performance reports.

**Objectives:**

* **Streamline Submission Processes**: Simplify how students submit assignments.
* **Enhance Teacher Efficiency**: Reduce administrative tasks through automation.
* **Provide Analytics**: Offer insights into student performance to inform teaching strategies.

**Benefits:**

* **Time Savings**: Automation saves time for both students and faculty.
* **Improved Accountability**: Students can easily track deadlines and submissions.
* **Data Management**: Simplifies retrieval and analysis of performance data.

**LabTracker** aligns with the Computer Science Department's goals by integrating technology into education, preparing students for industry practices.

## 1.5 References

* IEEE 830-1998: IEEE Recommended Practice for Software Requirements Specifications.
* Django Documentation: <https://docs.djangoproject.com>
* GitHub API Documentation: <https://docs.github.com/en/rest>
* Project Management Institute (PMI): Standards and guidelines for project management practices.

# 2. Overall Description

## 2.1 Product Perspective

**LabTracker** is an enhancement to traditional lab management systems specifically tailored for the Department of Computer Science at AMU. While existing systems manage lab assignments and track submissions, they lack the ability to integrate directly with GitHub, which is a vital tool for modern software development.

**Context and Origin:**

* **Motivation**: With the rise of GitHub as a standard for code collaboration, a solution that automates submission tracking and assessment is essential.
* **Target Audience**: Designed for students and faculty in the Computer Science department, LabTracker streamlines the submission process and provides insights into student performance.
* **Unique Positioning**: LabTracker distinguishes itself by automating the retrieval of assignments and tracking progress via GitHub, improving efficiency and accuracy.

**Relation to Larger Systems:**

LabTracker serves as an integral component that enhances existing lab management systems through GitHub integration, connecting to:

* **GitHub**: For real-time submission tracking.
* **Existing Lab Management Systems**: Future versions may offer interoperability with other educational platforms.

## 2.2 Product Features

**LabTracker** provides several key features that enhance lab management by integrating with GitHub to automate tracking of student assignments. The major features are organized as follows:

**1. Student Management**

* **Student Registration**: Enables students to sign up and create profiles within the system.
* **GitHub Profile Validation**: Verifies the existence of a student’s GitHub account and checks for the necessary repositories related to their courses.

**2. Assignment Tracking**

* **GitHub Submission Monitoring**: Automatically tracks student submissions by monitoring their GitHub repositories for commit dates and activity related to assigned problems.
* **Completion Status Tracking**: Updates the completion status of assignments based on the last commit date from GitHub.

**3. Reporting and Analytics**

* **Individual Student Reports**: Generates detailed reports for faculties on individual student performance, including submission dates and problem completion statuses.
* **Class Performance Reports**: Provides aggregate analytics for the entire class, highlighting trends in submissions, on-time completions.
* **Faculty Activity Logs**: Maintains records of faculty actions within the system, such as problem creation and deadline settings.

**4. Faculty Management**

* **Assignment Creation**: Allows faculty members to create and manage programming problems, including descriptions and deadlines.
* **Deadline Management**: Enables faculty to set and update deadlines for assignments linked to specific problems.