

A Project Report on

**SmartSubmit -AI-Powered Report  
Submission and Collaboration  
Platform**

Submitted by

**Bhargavi Deshmukh (Roll No:BT31034 )**  
**Shrutika Ghuge (Roll No:BT31049)**  
**Yash Gomte (Roll No:BT31050)**  
**Vaishali Jagtap(Roll No:BT31064)**

Under the guidance of

**Ms. B. P. Chaudhari**

In partial fulfillment of the award of Bachelor of Technology in Computer Science  
and Engineering



**Department of Computer Science and Engineering**  
**Maharashtra Institute of Technology,**  
**Chh. Sambhajinagar (Aurangabad)**  
**[2024-25]**

## **DECLARATION**

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Chh. Sambhajinagar (Aurangabad).

Date:

**Bhargavi Deshmukh (Roll No:BT31034 )**

**Shrutika Ghuge (Roll No:BT31049)**

**Yash Gomte (Roll No:BT31050)**

**Vaishali Jagtap(Roll No:BT31064)**

## **CERTIFICATE**

This is to certify that the major project- I report entitled “**SmartSubmit - AI-Powered Report Submission and Collaboration Platform**”, submitted by **Bhargavi Deshmukh, Shrutika Ghuge , Yash Gomte , Vaishali Jagtap** is the bonafied work completed under my supervision and guidance in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering, Maharashtra Institute of Technology under Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar (Aurangabad).

Place: Chh. Sambhajinagar (Aurangabad).

Date:

**Ms. B. P. Chaudhari**  
Guide

**Dr. S. L. Kasar**  
Head of Department

**Dr. N. G. Patil**  
Director  
Maharashtra Institute of Technology  
Chh. Sambhajinagar (Aurangabad).

## APPROVAL CERTIFICATE

This major project-I report entitled “**SmartSubmit - AI-Powered Report Submission and Collaboration Platform**” by **Bhargavi Deshmukh , Shrutika Ghuge , Yash Gomte , Vaishali Jagtap** is approved for Bachelor of Technology in Computer Science and Engineering, Maharashtra Institute of Technology under Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar (Aurangabad).

Place: Chh. Sambhajinagar (Aurangabad).

Date:

Examiner: \_\_\_\_\_  
(Signature)

\_\_\_\_\_  
(Name)

# INDEX/CONTENTS

<b>TITLE</b>	<b>PAGE NO</b>
Declaration	
Certificate	
Acknowledgement	
Approval certificate	
List of Figures	<b>i</b>
Abstract	<b>ii</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Necessity	1
1.2 Problem definition	2
1.3 Objectives	2
1.4 Scope and limitations	3
1.5 Applications	3
1.6 Organization and project plan	4
<b>2. LITERATURE SURVEY</b>	<b>10</b>
2.1 Overall Description	10
2.2 Project Perspective	11
2.3 Key Features	12
2.4 Future Scope	13
<b>3. SYSTEM DESIGN</b>	<b>15</b>
3.1 Proposed System	15
3.2 System Architecture	18
3.3 Database Specification	19
3.4 Algorithms, Techniques & Procedures	21
3.5 Hardware and Software	23
<b>REFERENCE</b>	

## List of Figures

<b>Figure</b>	<b>Illustration</b>	<b>Page</b>
1.1	Project Workflow Diagram	6
3.1	system flow diagram	16
3.2	System Architecture Overview	18
3.3	Use Case Diagram	21

## **ABSTRACT**

The digital transformation of education has necessitated the development of intelligent systems to manage academic submissions efficiently. Smart Submit is an AI-enhanced web-based platform aimed at automating and simplifying the student report submission and evaluation process. This system enables students to upload project or assignment reports, which are immediately processed through AI modules for grammar checking and content relevance analysis. The results are presented to faculty members, who can then provide additional comments, approve the submission, or request revisions. The platform supports real-time collaboration through integrated chat and live comment features, significantly enhancing communication between students and teachers. Smart Submit incorporates a modular architecture consisting of a robust backend server handling authentication, feedback processing, notifications, analytics, and integration services. The frontend interface provides easy-to-use features for user management, submission handling, and evaluation interfaces. The system is supported by a structured database that stores user profiles, submission metadata, deadlines, and integrates with external Learning Management Systems (LMS). It ensures data security, improves transparency in academic reviews, and promotes timely feedback loops. The project demonstrates the effective use of modern web technologies, RESTful APIs, and machine learning techniques to address inefficiencies in traditional submission workflows. By automating mundane tasks and enhancing academic communication, Smart Submit serves as a scalable solution for educational institutions looking to digitize and modernize their assessment systems.

# 1. INTRODUCTION

In modern educational environments, the submission and evaluation of academic reports remain one of the most critical components of continuous assessment. With the growing number of students and the volume of assignments being submitted, educators face increasing challenges in managing deadlines, reviewing reports thoroughly, and providing timely feedback. Traditional manual methods of submission and evaluation are time consuming, error-prone, and inefficient in large-scale academic settings. To address these issues, the integration of digital technologies and artificial intelligence (AI) has emerged as a transformative solution. The proposed system, SmartSubmit, is a web-based platform designed to streamline the entire academic report submission lifecycle—from student submission to automated grammar and formatting checks, teacher review, and approval or resubmission. By incorporating AI-driven modules and real-time collaboration features, SmartSubmit enhances the efficiency, transparency, and quality of academic evaluations. The system allows students to upload reports through a user-friendly interface while ensuring that each submission undergoes automated checks for grammatical accuracy and content relevance. Teachers can review AI feedback, add comments, and either approve or request modifications. Furthermore, automated notifications and deadline tracking enhance communication and accountability for both students and faculty. This project aims to bridge the gap between modern educational needs and existing systems by introducing an intelligent, scalable, and easy-to-use platform that supports improved academic workflows.

## 1.1 Necessity

The traditional method of academic report submission is riddled with inefficiencies and challenges. Students often face last-minute formatting issues, unclear submission processes, and a lack of constructive feedback. Faculty members spend a considerable time reviewing repetitive grammatical errors and structural flaws, leaving little room for providing meaningful guidance.

Moreover, existing tools like email, Google Docs, and WhatsApp are fragmented and do not offer an integrated solution for submission, evaluation, and feedback. These tools lack AI-driven support for improving writing quality and do not facilitate seamless communication between students and faculty. The environmental impact of



paper-based submissions is another growing concern. Institutions still relying on printed reports contribute significantly to paper waste, contrary to global sustainability efforts. Smart Submit addresses these challenges by offering an AI-powered platform that:

- Automates grammar and structure checks,
- Streamlines the submission-review cycle,
- Encourages interactive feedback through real-time collaboration,
- Promotes paperless digital submissions.

This centralized, intelligent solution is necessary to enhance the academic workflow, improve student writing skills, and support sustainability goals.[1]

## **1.2 Problem definition**

In academic institutions, the manual collection, organization, and evaluation of student assignment submissions can be highly inefficient and error-prone. Traditional methods often lead to issues such as misplaced submissions, delayed evaluations, and lack of a centralized system for tracking student performance. Moreover, as the volume of students and assignments grows, it becomes increasingly difficult for educators to manage the submission process effectively.

The Smart Submit project aims to address these challenges by developing a digital platform that allows students to submit their assignments online and enables educators to manage, evaluate, and track submissions efficiently. By automating and streamlining the submission workflow, the system seeks to reduce administrative overhead, ensure timely submissions, and maintain a structured repository of all academic tasks.

## **1.3 Objectives**

The primary objectives of the Smart Submit project are as follows:

1. To develop a centralized platform for the online submission of assignments by students.
2. To streamline the assignment management process for faculty members, enabling efficient review and feedback.
3. To ensure timely submission tracking, reducing chances of late or missed assignments.

4. To maintain a secure and organized repository of all submitted academic tasks for easy access and future reference.
5. To implement user roles and permissions, ensuring that only authorized users can perform specific actions (e.g., students submit, teachers review).
6. To enhance transparency and communication between students and educators regarding assignment status and deadlines.
7. To minimize manual work and errors, thereby improving the overall academic workflow.

## **1.4 Scope and Limitations**

### **1.4.1 Scope**

The Smart Submit project is designed to serve as a digital platform for managing assignment submissions in academic institutions. The system will support student registration, assignment uploads, deadline tracking, and faculty access for reviewing and grading submissions. It will provide role-based access for students and faculty, secure storage of submitted files, and a user-friendly interface for interaction.

The platform focuses on streamlining the submission process and making assignment handling more efficient and transparent. It is intended for use within educational institutions such as schools, colleges, and universities.[2]

### **1.4.2 Limitations:**

- The system currently supports only text-based and document-based assignment multimedia submissions are not supported.
- Evaluation and grading are manually done by faculty and not automated.
- The platform requires internet connectivity for access and operation.
- Integration with external Learning Management Systems (LMS) is not included in the initial version.
- User authentication is basic and may need enhancement for higher-level institutional deployment.

## **1.5 Applications**

The Smart Submit platform can be applied in various educational and institutional contexts where assignment management and submission tracking are essential. The

system enhances efficiency, transparency, and accessibility in academic workflows.

Below are some key applications:

- **Educational Institutions**

Schools, colleges, and universities can use Smart Submit to streamline assignment submission, review, and feedback processes.

It ensures timely submissions and helps faculty manage large volumes of student work effectively.

- **Online Learning Platforms**

E-learning platforms and virtual classrooms can integrate Smart Submit for organized assignment tracking and evaluation.

- **Corporate Training Programs**

Organizations offering employee training or skill development courses can use the system for managing project submissions and assessments.

- **Remote Education**

In situations where students and educators are geographically separated, Smart Submit provides a centralized digital platform for managing assignments efficiently.

- **Departmental Workflows**

Can be adapted for use in research labs, academic projects, and internal department submissions where structured documentation and deadline tracking are important.

## **1.6 Organization and project plan**

### **1.6.1 Organization of the Project:**

The Smart Submit project is organized in a structured manner to ensure smooth collaboration, timely delivery, and high-quality outcomes. Each team member is assigned specific roles and responsibilities based on their skill sets, and communication among team members is maintained through regular meetings and progress updates. The organization of the project is as follows:

**1. Team Roles and Responsibilities:** Project Manager: Responsible for overall coordination, task assignment, deadline tracking, and communication with stakeholders. Ensures that each phase of the project is completed on time.

**-Frontend Developers:** Design and implement the user interface using HTML, CSS, and JavaScript. Ensure that the application is user-friendly and responsive.

**-Backend Developers:** Develop APIs, handle database interactions, and manage server-side logic using Django. Ensure security, validation, and efficient data handling.

**-Database Administrator (DBA):** Designs the database schema and ensures data integrity and backup strategies. Handles performance tuning and SQL optimization.

**-Quality Assurance (QA) and Testing:** Conducts unit testing, integration testing, and user acceptance testing (UAT). Identifies bugs and ensures the system meets functional and non-functional requirements.

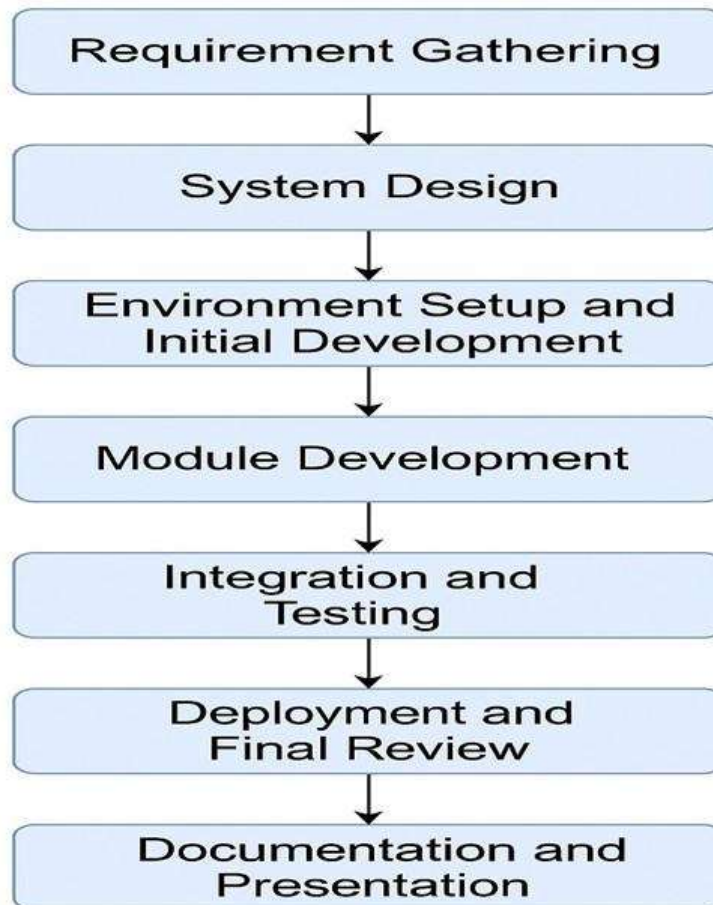
**-Documentation and Presentation:** Prepares technical documentation, user manuals, and project reports. Designs final presentations and ensures proper formatting of submission materials.

**2. Team Communication:** Weekly team meetings are scheduled to discuss progress, roadblocks, and next steps. Task management tools (e.g., Trello, Notion, or Google Sheets) are used to assign and track tasks. Communication tools such as WhatsApp or email are used for quick updates and coordination.

**3. Development Methodology:** The project follows a modular and incremental development approach. Each feature is developed independently, tested, and then integrated into the main system. Feedback from testing is incorporated into the next development cycle to improve functionality.

**4. Version Control:** A Git-based version control system (e.g., GitHub or GitLab) is used to track changes in the codebase. Branches are created for new features, and pull requests are reviewed before merging.

**5. Review and Feedback:** Regular code reviews and feedback sessions are held to ensure code quality and team learning. Faculty or mentor feedback is gathered at different stages for continuous improvement.



**Fig. 1.1: Project Workflow Diagram for Smart Submit**

### **1.6.2 Project plan**

The development of the Smart Submit system is structured into well-defined phases, each with specific goals, deliverables, and timelines. The project follows a phased development model combined with elements of agile methodology, allowing for iterative improvements based on feedback and testing.

Additional key considerations in the project plan include:

**Risk Assessment and Mitigation:** At each phase, potential technical or resource-related risks are identified and strategies are outlined to mitigate them, such as backup plans, alternate tools, or buffer time in the schedule.

**Feedback and Evaluation Cycles:** After each major milestone (e.g., system design, module development), peer or mentor reviews are conducted to ensure alignment with project objectives and quality standards.

**Progress Tracking:** Tools like Gantt charts and project management software (e.g., Trello or GitHub Issues) are used to monitor task completion, deadlines, and team responsibilities in real-time.

**Iterative Deployment:** Instead of deploying the system all at once, each functional module is tested and integrated gradually, ensuring better control over bugs and performance.

**Stakeholder Communication:** Regular updates are scheduled with faculty mentors and stakeholders to demonstrate progress, gather input, and adapt plans if necessary.

**Resource Allocation:** Hardware, software, and team resources are carefully assigned based on skill sets and project needs, ensuring optimal use of time and tools.

**Documentation at Every Stage:** From requirement analysis to testing, detailed documentation is maintained for each phase to support transparency, maintainability, and ease of future enhancement.

These additions help ensure the project is not only technically sound but also efficiently managed, adaptable, and aligned with academic goals.[3]

To ensure smooth execution and minimize risks, each phase of the Smart Submit project will include continuous monitoring and evaluation checkpoints. Weekly team meetings will be conducted to assess progress, identify and resolve bottlenecks, and revise development goals where needed. Task assignments and deadlines will be managed using project tracking tools such as Gantt charts and digital task boards like Trello or GitHub Projects to ensure accountability. A feedback loop involving sample users (students and faculty) will be incorporated after key milestones to validate the system's usability, user interface, and practical functionality.

Moreover, an iterative testing approach will be followed, where each module undergoes unit and integration testing immediately after development. This prevents error accumulation and allows for early bug detection. Special attention will be paid to scalability, security, and user experience during the implementation phase. To avoid disruptions, risk mitigation strategies like regular data backups, secure version control (using Git), and setting up fallback environments will be enforced.

Additionally, the project will include mid-phase documentation to record design decisions, changes, and challenges for future reference. Peer code reviews will be organized before merging critical modules to ensure coding standards and quality. At the end of each phase, performance will be evaluated against predefined KPIs (Key Performance Indicators) to ensure the project stays aligned with its goals and timeline.

### **Phase 1: Requirement Gathering (Week 1)**

Conduct brainstorming sessions to understand the needs of both students and faculty.

Identify core features such as user roles, assignment submission, and review systems.

Document functional and non-functional requirements.

Finalize the project scope, goals, and limitations.

Allocate roles and responsibilities among team members.

### **Phase 2: System Design (Week 2)**

Create high-level architecture diagrams and data flow diagrams (DFD).

Design the database schema with entities such as users, assignments, and submissions.

Create UI wireframes or mockups for key pages (login, dashboard, upload, etc.).

Choose the appropriate tech stack (e.g., Django, PostgreSQL, HTML/CSS).

### **Phase 3: Environment Setup and Initial Development (Weeks 3–4)**

Set up the development environment with Django framework and database.

Implement basic user authentication system (login, register, logout).

Set up version control (GitHub/GitLab) for collaborative coding.

Begin frontend development: login page, student/faculty dashboard layout.

### **Phase 4: Module Development (Weeks 5–6)**

Develop key features:

- i. Assignment creation (by faculty)
- ii. Submission module (for students)
- iii. Review and feedback (by faculty)
- v. File upload and download

Connect frontend to backend using Django views and templates (or APIs if

Django Ninja DRF)

Implement deadline logic and validation rules.

Build admin panel for user and assignment management.

### **Phase 5: Integration and Testing (Week 7)**

Integrate all modules and test the complete flow from registration to submission .

Perform different levels of testing: Unit Testing (individual components smarter

Integration Testing (end-to-end workflow) User Acceptance Testing (with sample data) Fix bugs and improve UI/UX based on feedback.

### **Phase 6: Deployment and Final Review (Week 8)**

Host the project on a local server or cloud platform (e.g., Heroku, PythonAnywhere).

Ensure database backup and system security.

Collect feedback from test users (students/faculty) for final tweaks.

Conduct final code review and performance testing.

### **Phase 7: Documentation and Presentation (Week 9)**

Prepare the final project report with all sections (problem definition, objectives, architecture, etc.).

Create a presentation for viva or demonstration, including screenshots and demo video (if needed).

Finalize user manual and help guide.



## **2. LITERATURE SURVEY**

### **2.1 Overall description**

To design a robust and user-friendly project submission and evaluation platform, a review of existing systems such as Turnitin, Moodle, and Google Classroom was conducted. The table below summarizes key features, advantages, and drawbacks of these systems, which helped identify the gaps addressed by Smart Submit. In recent years, the digital transformation of academic workflows has seen significant developments, particularly in areas such as online submissions, plagiarism detection, and automated evaluation. Tools like Turnitin have become standard in universities worldwide, offering plagiarism checking and basic feedback features. Turnitin allows document submission and mentor reviews, but its limitations lie in the absence of real-time collaboration, in-depth formatting validation, and detailed grammar analysis, which restrict its use for holistic academic evaluation.

Moodle, another well-known open-source Learning Management System (LMS), supports various educational activities such as assignment uploads, grading, and forums. It is highly customizable and adaptable, but lacks built-in AI functionalities like automatic grammar checks or report formatting validation. Its feedback mechanism is also relatively static, with limited support for inline, context-aware suggestions.

Google Classroom offers a cloud-based approach to assignment handling with seamless integration with Google Drive and Docs. While it is appreciated for its ease of use, mobile access, and quick setup, it doesn't support version tracking or deep academic analysis tools such as plagiarism and grammar detection. Teachers have limited tools for assessing report formatting or content structure beyond manual checks.

On the other hand, enterprise-level platforms like Blackboard Learn and Canvas LMS provide comprehensive learning environments with features like course management, detailed analytics, and assignment tracking. Blackboard supports discussion boards, extensive grading tools, and is scalable for large institutions, but suffers from a complex interface and high licensing costs. Canvas offers a modern UI and strong third-party 15 integration capabilities, but sometimes faces synchronization issues and lacks strong offline access features, which can hinder seamless academic operations.

Despite the strengths of these platforms, a key limitation across all of them is the lack of automated formatting verification, grammar correction, intelligent feedback loops, and streamlined resubmission workflows. These gaps highlight the need for a dedicated solution that caters specifically to the challenges faced during academic report evaluations. SmartSubmit is designed to fill this gap by offering a unified platform where students can submit academic reports that undergo automated grammar and formatting checks before reaching the faculty for evaluation. It introduces features such as AI-enhanced text analysis, structured resubmission workflows, role-based access control, and deadline tracking to support continuous improvement and reduce manual workload. The system is built using open-source technologies to ensure affordability and scalability, making it suitable for educational institutions of all sizes.

Additionally, SmartSubmit fosters transparency and efficiency in the evaluation process, ensuring students receive timely, actionable feedback while allowing educators to maintain high academic standards with reduced administrative burden. The literature thus emphasizes the necessity and relevance of a solution like SmartSubmit in modern academic settings where quality, integrity, and timely feedback are paramount. [4]

## **2.2 Project perspective**

The Smart Submit project aims to provide a modern, secure, and user-friendly platform for students and instructors to manage the submission and evaluation of academic work. Unlike traditional submission platforms, Smart Submit focuses on seamless integration of deadlines, version control, plagiarism detection, and real time collaboration features. While existing systems such as Turnitin, Google Classroom, and Moodle provide basic submission and feedback functionalities, they often lack comprehensive deadline management, advanced feedback mechanisms, and real-time collaboration for iterative improvements. By addressing these gaps, Smart Submit will provide an enhanced user experience, supporting the efficient management of academic submissions and reducing common issues faced by both students and instructors, such as missed deadlines and poor tracking of submission revisions.

This project builds upon existing systems by enhancing their core functionalities, enabling a smarter, more integrated submission platform for educational institutions.

The Smart Submit platform will serve as a scalable solution for both small and large academic organizations, with a focus on simplicity and effectiveness in submission management. [6]

### **2.3 Key Features**

The Smart Submit platform incorporates several key features designed to enhance the submission and evaluation process for students and instructors. These features aim to address gaps identified in the literature survey and provide a more streamlined, efficient, and user-friendly experience. The key features of Smart Submit are as follows:

#### **1. Real-time Submission Tracking**

Students and instructors can track submission progress, ensuring that all stages of the submission process are completed on time. This feature helps manage deadlines and avoids last-minute rushes.

#### **2. Version Control**

Unlike traditional systems, Smart Submit allows users to track different versions of submitted documents. This ensures that changes are captured, making it easier to review progress and maintain a history of edits.

#### **3. Plagiarism Detection Integration**

Built-in plagiarism detection ensures the integrity of student submissions. It compares submitted work against multiple databases and provides comprehensive plagiarism reports to instructors.

#### **4. Automated Deadline Reminders**

The system automatically sends reminders to students about upcoming deadlines. Instructors also get notified of submission progress and can set custom deadlines for different stages of the assignment.

#### **5. Collaborative Feedback System**

Instructors can leave real-time feedback on student submissions, including inline comments and annotations. This feature promotes interaction and allows students to improve their work based on timely feedback.

#### **6. Role-based Access Control**

The platform supports different user roles, such as students, instructors, and administrators, each with access to specific features based on their permissions.

#### **7. Easy Integration with LMS Platforms**

Smart Submit integrates smoothly with popular Learning Management Systems (LMS) like Moodle and Google Classroom, allowing for easy assignment tracking, grading, and submission management.

## **8. Mobile and Desktop Support**

The platform is accessible on both mobile and desktop devices, ensuring that students and instructors can manage assignments and feedback on-the-go.[5]

## **2.4 Comparative Study**

The Smart Submit platform has been designed with a scalable and extensible architecture, allowing numerous opportunities for future enhancements. One of the major future developments includes the creation of a mobile application for Android and iOS devices, enabling students and instructors to manage submissions, feedback, and notifications conveniently through their smartphones. Additionally, the system can be enhanced by integrating advanced AI-based evaluation techniques such as automatic grading of descriptive answers, intelligent content feedback, and predictive analytics to track student performance trends. Integration with popular Learning Management Systems (LMS) like Moodle, Google Classroom, and Canvas would further streamline the assignment management process across platforms. Multi-language support can also be incorporated, making the system accessible to a wider and more diverse range of users. Another potential upgrade involves introducing an offline submission feature, allowing students to upload assignments without an active internet connection and syncing them later once connected.

The Smart Submit platform is designed with scalability and continuous improvement in mind. Although the current version focuses on core submission and evaluation functionalities, there is significant potential to expand its capabilities in the future. The following are some of the key areas for future development:

### **1. AI-Based Feedback Suggestions**

Integration of artificial intelligence to provide automated suggestions for grammar, structure, and content relevance, helping students improve their work before final submission.

### **2. Analytics Dashboard**

Implementation of detailed analytics for both students and instructors, offering insights such as submission trends, common errors, and performance tracking over time.

### **3. Offline Submission Support**

Development of features allowing students to draft or queue submissions offline, which are then uploaded automatically when internet connectivity is restored.

### **4. Multi-language Support**

Expanding the platform to support multiple languages for users from diverse linguistic backgrounds, making it globally adaptable.

### **5. Voice and Video Feedback**

Allowing instructors to give voice or video-based feedback, making evaluation more personalized and clear, especially for large project submissions or presentations.

### **6. Integration with Digital Certification Tools**

Automating certificate generation upon successful submission and evaluation of major assignments or projects, useful for MOOCs and online learning platforms.

### **7. Blockchain-based Submission Authentication**

Leveraging blockchain technology to ensure authenticity, prevent tampering, and add a layer of security for high-value submissions.

### **8. Research Collaboration Module**

Adding a collaborative research space for students and guides to work on long term research projects with document sharing, progress tracking, and version syncing.[7]

### 3. SYSTEM DESIGN

#### 3.1 Proposed system

The Smart Submit system is designed to streamline the project and assignment submission process between students and instructors, ensuring efficiency, transparency, and security at every step.

The proposed system addresses the limitations of existing platforms by offering real time tracking, version control, deadline management, plagiarism detection, and collaborative feedback features within a unified platform.

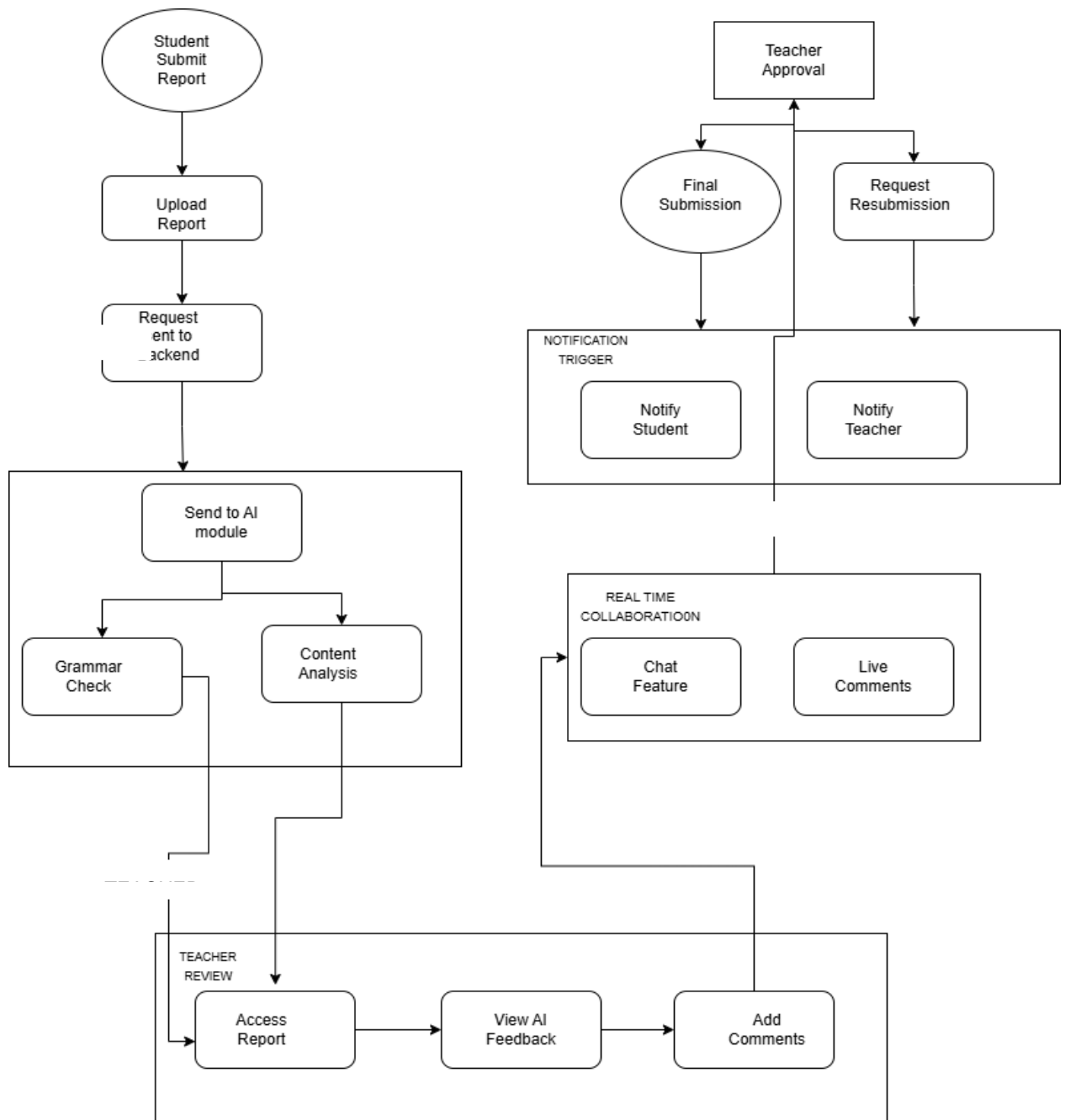
The core concept behind Smart Submit is to create a centralized portal where students can submit their work, instructors can provide feedback and evaluations, and both parties can track the progress throughout multiple iterations. The system is built to handle individual and group submissions, manage deadlines for different stages of work (like drafts, final reports, presentations), and ensure submission authenticity through plagiarism scanning.

Smart Submit will include modules for:

- **User Authentication and Authorization:** Secure login and role-based access control for students, instructors, and admins.
- **Submission Management:** Allowing students to upload, update, and track assignment submissions.
- **Deadline and Reminder System:** Setting deadlines with automatic reminders and notifications.
- **Version Control:** Managing multiple versions of submitted documents for better tracking.
- **Feedback and Evaluation:** Enabling instructors to review submissions, add inline comments, and provide structured feedback.
- **Plagiarism Detection:** Integrating plagiarism scanning tools to verify content originality.
- **Dashboard and Analytics:** Offering a graphical dashboard for students and instructors to monitor progress and submission status.

The Smart Submit system will thus act as a comprehensive, user-centric platform catering to the needs of modern academic institutions, promoting better engagement, higher submission quality, and efficient evaluation workflows. The Smart Submit

system is designed as a comprehensive platform to handle project submissions, track deadlines, detect plagiarism, and facilitate real-time feedback between students and instructors. The system aims to overcome the limitations of existing platforms by integrating modern features into a centralized, user-friendly portal.



**Fig. 3.1 System flow diagram**

### **High-Level and Low-Level Design Description:**

The system workflow for Smart Submit starts with the student submitting a report. The report is uploaded and a request is sent to the backend for further processing.

At the backend, the system enters the AI Processing module, where the report is sent to an AI module that performs two major analyses:

**Grammar Check:** Identifies grammatical errors in the submitted report.

**Content Analysis:** Analyzes the overall quality, relevance, and originality of the content. After AI analysis, the system allows students and teachers to access the report and view AI-generated. Feedback Teachers can then add manual comments for further improvements or suggestions. In parallel, the system supports Real-Time Collaboration, enabling:

**Chat Feature:** Direct communication between students and teachers.

**Live Comments:** Instant commenting on specific sections of the report.

Post review and feedback, the teacher makes a decision:

**Final Submission Approval:** If the report meets the required standard.

**Request Resubmission:** If corrections or improvements are needed.

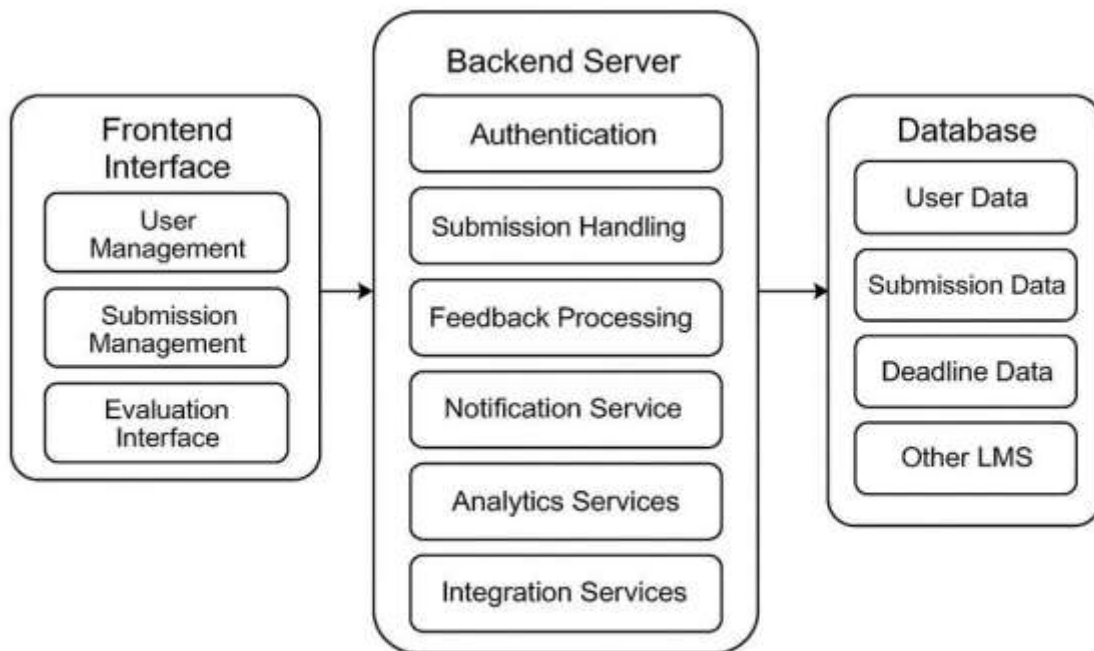
Whenever a significant action occurs (approval or resubmission request), the Notification Trigger module ensures that both students and teachers are notified immediately to maintain quick communication.

Thus, the Smart Submit system seamlessly combines AI evaluation, real-time collaboration, and manual teacher intervention to create an effective, user-friendly academic report management platform.

- **Version Control:** Maintains a history of all revisions and comments for traceability.
- **Deadline and Status Tracking:** Each report carries a timestamp, current status (Draft, Submitted, Under Review, Approved, Returned), and countdown to the due date.
- **Notification Trigger Module:** Sends automated alerts via email or in-app messages for every significant action—submission received, feedback added, approval granted, or resubmission requested.



### 3.2 System architecture



**Fig. 3.2 System Architecture Overview**

#### **System Architecture Description:**

The Smart Submit system architecture is divided into three major layers, ensuring smooth interaction, efficient processing, and secure data management:

##### **1. Frontend Interface:**

Handles all user interactions including user registration, login, submission uploads, and evaluation views.

Provides an intuitive and user-friendly interface for both students and teachers.

Supports responsive design for access across multiple devices (desktop, table).

##### **2. Backend Server:**

Acts as the core processing unit managing business logic and server-side operations.

Handles secure authentication, submission storage, and feedback generation.

Processes notifications for important events like submission deadlines or backless feedback availability.

Provides analytics services to track submission trends and performance statistics.

Supports integration with external Learning Management Systems for broader for compatibility.

### **3. Database:**

Stores structured data such as user profiles, submitted assignments, feedback comments, and deadlines.

Ensures data integrity and security through robust database management practices.

Supports future scalability to integrate with other educational platforms and services.

The modular three-tier architecture of Smart Submit optimizes system performance by separating concerns across the frontend, backend, and database layers. This separation allows each layer to scale independently frontend servers can be replicated to handle high user concurrency, backend services can be containerized for efficient load balancing, and databases can be tuned with indexing strategies for rapid query responses. Compared to monolithic, single-tier systems, this design reduces bottlenecks, enhances maintainability, and ensures that critical components such as AI processing and notification services operate with minimal latency under peak workloads.

### **3.3 Database Specification**

The Smart Submit platform uses a relational database to efficiently store and manage user data, assignments, submissions, feedback, and notifications. The database is structured into several key tables to ensure smooth operation and maintain data integrity.

**1. Users Table:** This table stores the information of all users on the platform, including students, instructors, and administrators. Each user is assigned a unique `user_id` and their role (student, instructor, or admin) is recorded. The table also stores personal details such as name, email, and `password_hash` for secure authentication. Additionally, the timestamps `created_at` and `updated_at` track when the user account was created and last updated.

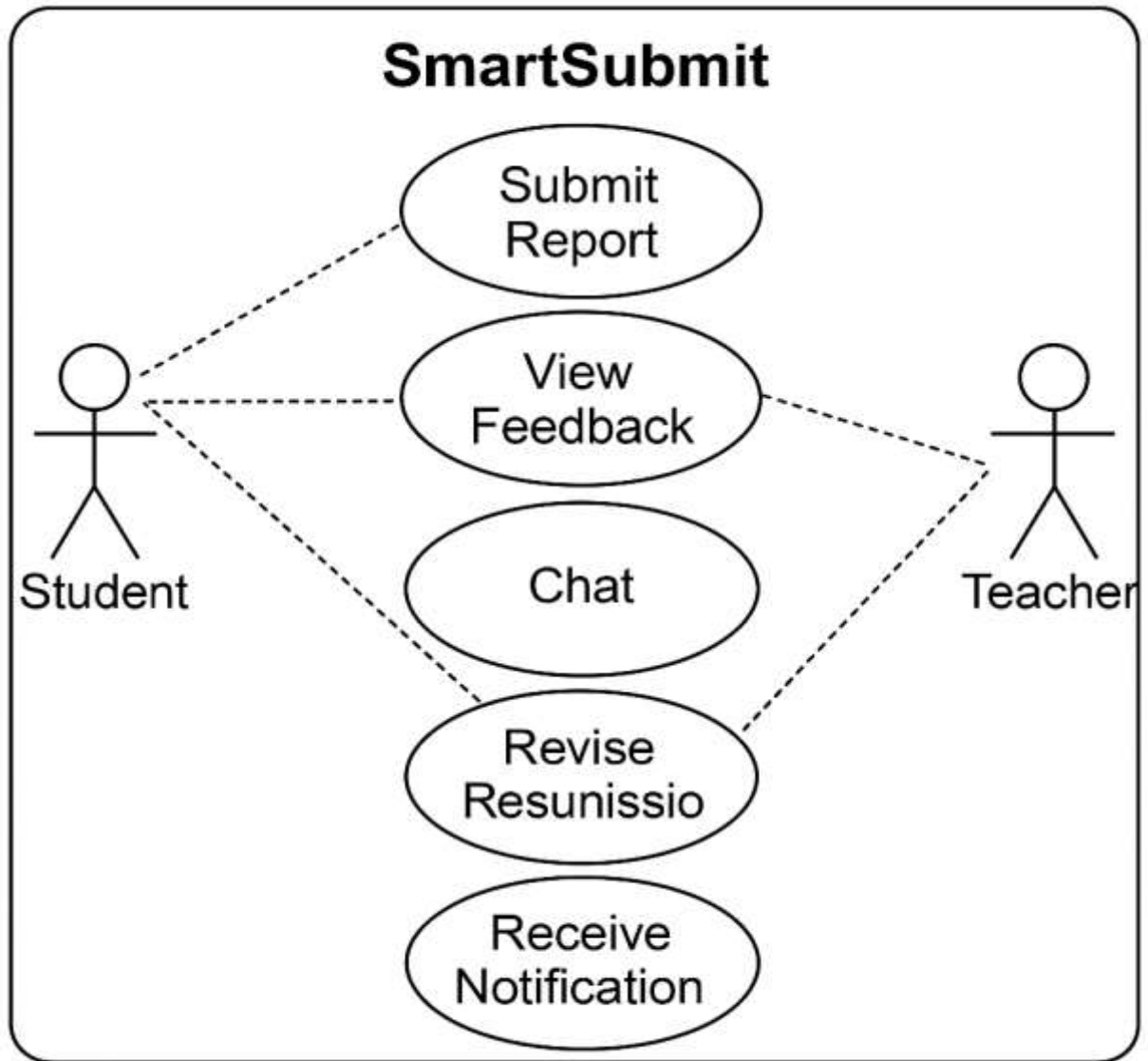
**1. Assignments Table:** The Assignments Table holds details about assignments created by instructors. Each assignment has a unique `assignment_id` and contains

information such as the title, description, and due\_date. The instructor\_id links back to the Users table to associate the assignment with the instructor who created it.

**2. Submissions Table:** This table stores the submissions made by students for assignments. Each submission is linked to a student through the user\_id and an assignment through the assignment\_id. The file\_path column contains the location of the uploaded file, while version tracks different versions of the same submission. The submission\_date column records when the submission was made, and the plagiarism\_score column holds the result of any plagiarism check performed on the submission.

**2. Feedback Table:** The Feedback Table stores the feedback provided by instructors for each submission. Each feedback entry is tied to a specific submission via submission\_id. The instructor\_id column links to the instructor providing the feedback, and the comments column holds the instructor's remarks on the submission. The grade column records the evaluation, and feedback\_date indicates when the feedback was given.

**3. Notifications Table:** This table is responsible for storing notifications sent to users. Each notification is associated with a specific user through the user\_id column, which indicates the recipient of the notification. The message column contains the notification content, and the status column keeps track of whether the notification has been read or not. Finally, date stores the timestamp when the notification was sent. The relationships between these tables ensure that data is stored in an organized manner, allowing for easy retrieval and updates. The Users table is linked to the Assignments, Submissions, Feedback, and Notifications tables via foreign keys, ensuring the system can accurately track users' actions and interactions. The database is designed to support scalability, enabling the platform to handle a growing number of users and assignments efficiently.



**Fig. 3.3 Use Case Diagram**

**3.4 Algorithms, Techniques & Procedures Used** In order to implement the Smart Submit platform efficiently and securely, the following key algorithms, techniques, and procedures have been employed:

**1. Authentication & Security**

**Password Hashing (bcrypt):** User passwords are hashed using the bcrypt algorithm with a work factor of 12, ensuring resistance to brute-force and rainbow-table attacks.

**Token-Based Authentication (JWT):** After login, users receive a JSON Web Token signed with an HMAC-SHA256 secret. Tokens carry role information (student, instructor, admin) and expire after one hour to limit exposure.

## **2. File Upload & Version Control**

**Chunked File Upload:** Large submissions are broken into fixed-size chunks, uploaded in parallel, and reassembled on the server to improve reliability and allow resume on failure. **Versioning Procedure:** Each upload increments a version counter. Metadata (version number, timestamp, previous version reference) is stored in the Submissions table. This enables rollback and side-by-side comparison of document revisions.

## **3. AI-Based Report Analysis**

**Grammar Checking (LanguageTool API):** Submitted text is sent to an opensource grammar-checking engine via REST calls; the JSON response of flagged issues is parsed and stored for inline display.

**Content Quality Analysis (TF-IDF & Classification):** The report text is vectorized using TF-IDF. A pre-trained Support Vector Machine (SVM) classifier then predicts a content-quality score (e.g., relevance to topic keywords), which helps instructors focus on low-quality sections.

## **4. Plagiarism Detection**

**Cosine-Similarity Algorithm:** The submitted document is split into n-grams ( $n=5$ ). TF-IDF vectors are computed for both the submission and a reference corpus; cosine similarity scores above a 0.7 threshold trigger a plagiarism alert.

**External API Integration:** Optionally, SmartSubmit can forward documents to third-party services (e.g., Turnitin) via secure API calls for deeper comparison against proprietary databases.

## **5. Notification & Scheduling**

**Asynchronous Task Queue (Celery + RabbitMQ):** Deadline reminders and feedback notifications are enqueued as Celery tasks. Workers process jobs in the background, sending emails or in-app alerts at scheduled times.

**Cron Fallback:** A nightly cron job scans for impending deadlines (24-hour window) and enqueues any missed notifications to ensure reliability.

## **6. Real-Time Collaboration**

**WebSocket Protocol (Django Channels):** Live chat and inline comment updates use WebSockets. A channel layer (Redis) broadcasts events to connected clients in real time, enabling synchronous discussion.

**Optimistic UI Updates:** Clients tentatively display new chat messages or comments immediately and reconcile against server acknowledgments to minimize perceived latency.

## **7. Analytics & Reporting**

**Database Indexing:** Frequently queried columns (e.g., `submission_date`, `user_id`, `assignment_id`) are indexed to accelerate dashboard queries.

**Aggregate SQL Queries:** Instructor dashboards use `GROUP BY` and window functions (e.g., `ROW_NUMBER()`) to compute submission counts, average delays, and plagiarism rates efficiently.

## **8. Data Integrity & Backup**

**Transactional Operations:** All critical sequences (e.g., version increment + file save + database insert) are wrapped in database transactions to prevent partial updates.

**Automated Backups:** Nightly database dumps and incremental file-storage snapshots are performed, with retention policies set to 30 days to support disaster recovery.

These combined algorithms and procedures ensure that SmartSubmit delivers a robust, secure, and high-performance platform for academic submission management.

### **3.5 Hardware and**

#### **Software Hardware**

##### **Requirements:**

**Processor:** Intel Core i5 or higher

**RAM:** Minimum 8 GB (Recommended 16 GB for server hosting)

**Hard Disk:** Minimum 500 GB storage

**Network:** Stable internet connection with minimum 10 Mbps bandwidth

**Backup Storage:** External drive or cloud storage (optional, for backup)

##### **Software Requirements:**

##### **Operating System:**

- i. Development Machine: Windows 10/11 or Ubuntu 20.04+
- ii. Server Hosting: Ubuntu Server 22.04 LTS recommended

##### **Programming Languages:**

- i. Python 3.11 or higher

**Frameworks & Libraries:**

- i. Django 5.0 (for backend development)
- ii. Django Ninja (for API development)
- iii. Django Channels (for real-time features)
- iv. Celery and Redis (for asynchronous task management)
- v. LanguageTool API (for grammar checking)

**Database:** i. PostgreSQL 15 (recommended) or MySQL 8.0

**Other Tools and Platforms:**

- i. Git (for version control)
- ii. Docker (for containerization, optional)
- iii. Postman (for API testing)
- iv. Visual Studio Code (for development environment)

**Browser Requirements:**

- i. Latest versions of Chrome, Firefox, or Edge

## REFERENCE

- [1] Zhai, C., & Massung, S. (2016). Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining. ACM Books.
- [2] Roscoe, R. D., & McNamara, D. S. (2013). Writing Pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105(4), 1010–1025.
- [3] D. Jurafsky and J. H. Martin, *Speech and Language Processing* 3rd ed. Draft, 2023. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
- [4] B. Pang, L. Lee, and S. Vaithyanathan. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the ACL- conference on Empirical methods in natural language processing*, pp. 79–86.
- [5] I. Anderson, "Plagiarism Detection Software: Promises and Pitfalls," *Educause Review*, vol. 47, no. 5, pp. 30–31, Sep. 2012.
- [6] S. B.Kotsiantis, "Use of Machine Learning Techniques for Educational Proposals," *Educational Technology & Society*, vol. 11, no. 2, pp. 39–52, 2008. M. D. Shermis and J. Burstein (Eds.), *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, Routledge, 2013.
- [7 ] Grammarly: Grammarly Inc. (<https://www.grammarly.com>) Overleaf: Overleaf by Digital Science (<https://www.overleaf.com>) Turnitin: Turnitin LLC (<https://www.turnitin.com>)



## ACKNOWLEDGEMENT

We would like to express special thanks to our beloved Guide **Ms. B. P. Chaudhari** for her valuable suggestion and rare insights, constant encouragement, and inspiration throughout the project.

We would like to take this opportunity to thank **Dr. S. L. Kasar, Head of the Department**, for providing her great support in the successful completion of our project.

We express our deep sense of gratitude to our beloved Principal, **Dr. N. G. Patil**, for his valuable guidance and for permitting us to carry out this project.

**Bhargavi Deshmukh (Roll No:BT31034)**

**Shrutika Ghuge (Roll No:BT31049)**

**Yash Gomte (Roll No:BT31050)**

**Vaishali Jagtap(Roll No:BT31064)**