

PARDARSH

Project report submitted in fulfillment of the requirement for the degree of
Bachelor of Technology

In

Computer Science and Engineering

By

SARVIN SHRIVASTAVA (2200290100144)

SATYAGYA SINGH (2200290100145)

SATYAM BALAIWAR (2200290100146)

SHASHWAT RAI (2200290100148)

Under the supervision of

Mr. Gaga Thakral (Assistant Professor, KIET)

To



KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

**KIET Group of Institutions, Delhi-NCR,
Ghaziabad (UP)**

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Sign:

Name: Sarvin Shrivastava

Roll No.: 2200290100144

Date: 12-05-2025

Sign:

Name: Satyam Balaiwar

Roll No.: 2200290100145

Date: 12-05-2025

Sign:

Name: Satyagya Singh

Roll No.: 2200290100146

Date: 12-05-2025

Sign:

Name: Shashwat Rai

Roll No.: 2200290100148

Date: 12-05-2025

CERTIFICATE

This is to certify that Project Report entitled "**Pardarsh**" which is submitted by Sarvin Shrivastava, Satyam Balaiwar, Satyagya Singh and Shashwat Rai in partial fulfillment of the requirement for the award of the Degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidate's own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date: 12-05-2025

Supervisor Name: Mr. Gaga Thakral
(Assistant Professor)

Table of contents

| Content | Page no. |
|-----------------------------|-----------------|
| Abstract | 4 |
| Chapter 1 | |
| Introduction | 5 |
| Problem Statement | 6 |
| Objective | 9 |
| Chapter 2 | |
| System Development | 12 |
| Functional Requirements | 16 |
| Non-Functional Requirements | 21 |
| References | 25 |

ABSTRACT

In today's era of heightened accountability and increasing demands for public trust, transparency in public procurement has become not just a regulatory requirement but a foundational pillar for fostering a more responsible and participatory governance ecosystem. Citizens expect more than just promises—they want visibility, traceability, and assurance that public resources are being utilized efficiently and ethically.

Pardarsh is a forward-looking, citizen-centric platform developed to bridge the information gap between government bodies, contractors, and the general public. It aims to empower both citizens and contractors by offering seamless, real-time access to updates on government-funded infrastructure and development projects. Going beyond the limitations of traditional tendering systems that typically focus on pre-award processes, Pardarsh emphasizes **post-award transparency**—an often overlooked but critically important phase in the project lifecycle.

The platform mandates contractors to regularly upload detailed updates on project milestones, execution status, financial disbursements, cost breakdowns, resource utilization, and acquisition-related documentation. This structured and publicly accessible data stream not only promotes accountability but also deters malpractices, reduces information asymmetry, and fosters a culture of proactive engagement between stakeholders. By creating a digital audit trail, Pardarsh enhances public oversight and builds long-term trust in governance mechanisms.

In essence, Pardarsh represents a significant step toward reimagining how public project information is disseminated, monitored, and consumed, ultimately contributing to a more transparent, efficient, and inclusive governance framework.

Chapter - 1

1.1 INTRODUCTION

The *Pardarsh E-Portal* represents a bold step toward fostering transparency and accountability in public procurement processes, a critical need in today's governance ecosystem. Public procurement is a cornerstone of government activity, involving significant financial investments to meet public infrastructure and service demands. However, the lack of adequate transparency and public involvement in monitoring these projects has led to concerns about inefficiency, corruption, and mismanagement. The *Pardarsh* project seeks to address these challenges by bridging the gap between contractors and the public, empowering citizens to oversee projects that directly affect their communities.

Unlike existing government portals that focus solely on the tendering process, *Pardarsh* centers on post-award transparency. Contractors are required to upload regular updates on project progress, including detailed cost breakdowns and acquisition documentation. This ensures that stakeholders have continuous visibility into how public funds are being utilized. Furthermore, a distinctive public rating system enables citizens to provide feedback on contractor performance. These ratings are aggregated and displayed on contractor profiles, serving as a measure of accountability and incentivizing quality work.

The platform's design is both user-friendly and robust, utilizing secure databases and dynamic interfaces to ensure seamless functionality. The use of structured data processing ensures that critical project information is presented clearly, making it accessible to individuals from all walks of life. By aligning with the principles of the Right to Information under the Indian Constitution, *Pardarsh* supports the democratic ideals of transparency and public participation.

1.2 PROBLEM STATEMENT

Public procurement is a vital process for governments, involving significant financial expenditure to provide infrastructure, services, and resources to the public. However, one of the most pressing issues in this domain is the lack of transparency and accountability, which often leads to inefficiency, mismanagement, and, in extreme cases, corruption. Reports suggest that a significant portion of global public procurement budgets—estimated to be around 10-30%—is lost to corruption and inefficiencies annually. In India, where infrastructure projects often face delays and cost overruns, public trust in the system has been eroded due to limited access to information on project execution and outcomes.

Existing government portals are primarily focused on the tendering process, providing platforms for applying and granting contracts. While these systems ensure a level of regulatory compliance in awarding contracts, they fail to offer mechanisms for continuous monitoring of project performance post-award. Citizens, who are the ultimate beneficiaries of these projects, often remain in the dark about how public funds are utilized, whether deadlines are met, or if contractors adhere to agreed standards. This lack of transparency not only hampers public trust but also reduces accountability among contractors.

Additionally, there is no efficient mechanism for citizens to provide feedback or hold contractors accountable. As a result, issues like delays, subpar work quality, and budget mismanagement often go unnoticed or unaddressed. This creates a gap in governance where contractors operate with minimal scrutiny, and public resources are at risk of being misused.

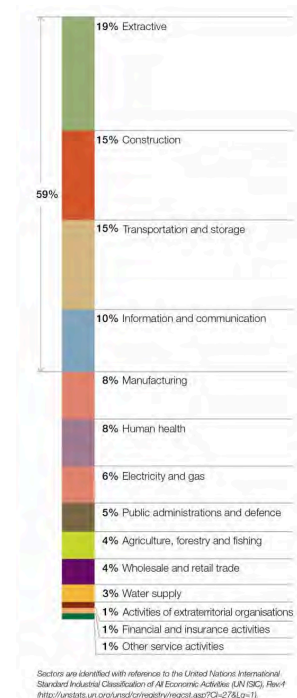


Fig 1

1.2.1 The high cost of corruption in public procurement

The direct costs of corruption include loss of public funds through misallocations or higher expenses and lower quality of goods, services and works (OECD,2015a). Those paying the bribes seek to recover their money by inflating prices, billing for work not performed, failing to meet contract standards, reducing quality of work or using inferior materials, in case of public procurement of works.

This results in exaggerated costs and a decrease in quality. A study by the OECD and the World Bank shows that corruption in the infrastructure and extractives sectors lead to misallocation of public funds and substandard and insufficient services (OECD, 2015a).

Although it is difficult to measure the exact cost of corruption due to its hidden nature, it has been estimated that between 10-30% of the investment in publicly funded construction projects may be lost through mismanagement and corruption (COST, 2012),and estimates of 20-30% of project value lost through corruption are widespread (Wells, 2014; Stansbury, 2005). The Construction Sector Transparency Initiative (CoST) also estimates that “annual losses in global construction through mismanagement,inefficiency and corruption could reach USD 2.5 trillion by 2020” (COST, 2012). Within theEuropean Union, corruption more generally is estimated to cost €120 billion per year (EuropeanCommission, 2014a), which represents approximately 1 % of the EU GDP and represented slightly less than the annual budget of the EU in 2014, which amounted to €143 billion (EuropeanCommission, 2014b).

In terms of indirect costs, corruption in public procurement leads to distortion of competition, limited market access and reduced business appetite for foreign investors. Not surprisingly, companies increasingly demand for improved fairness of public procurement procedures. The 2014 Business andIndustry Advisory Committee to the OECD (BIAC) Economic Survey indicates that enhancing efficiency and transparency in public procurement is the top priority for public sector reforms.

1.2.2 A myriad of integrity risks along the public procurement cycle

Integrity risks occur in every stage of the procurement process, from the needs assessment over the bidding phase to the contract execution and payment. conflict of interest, and various kinds of fraud risks.

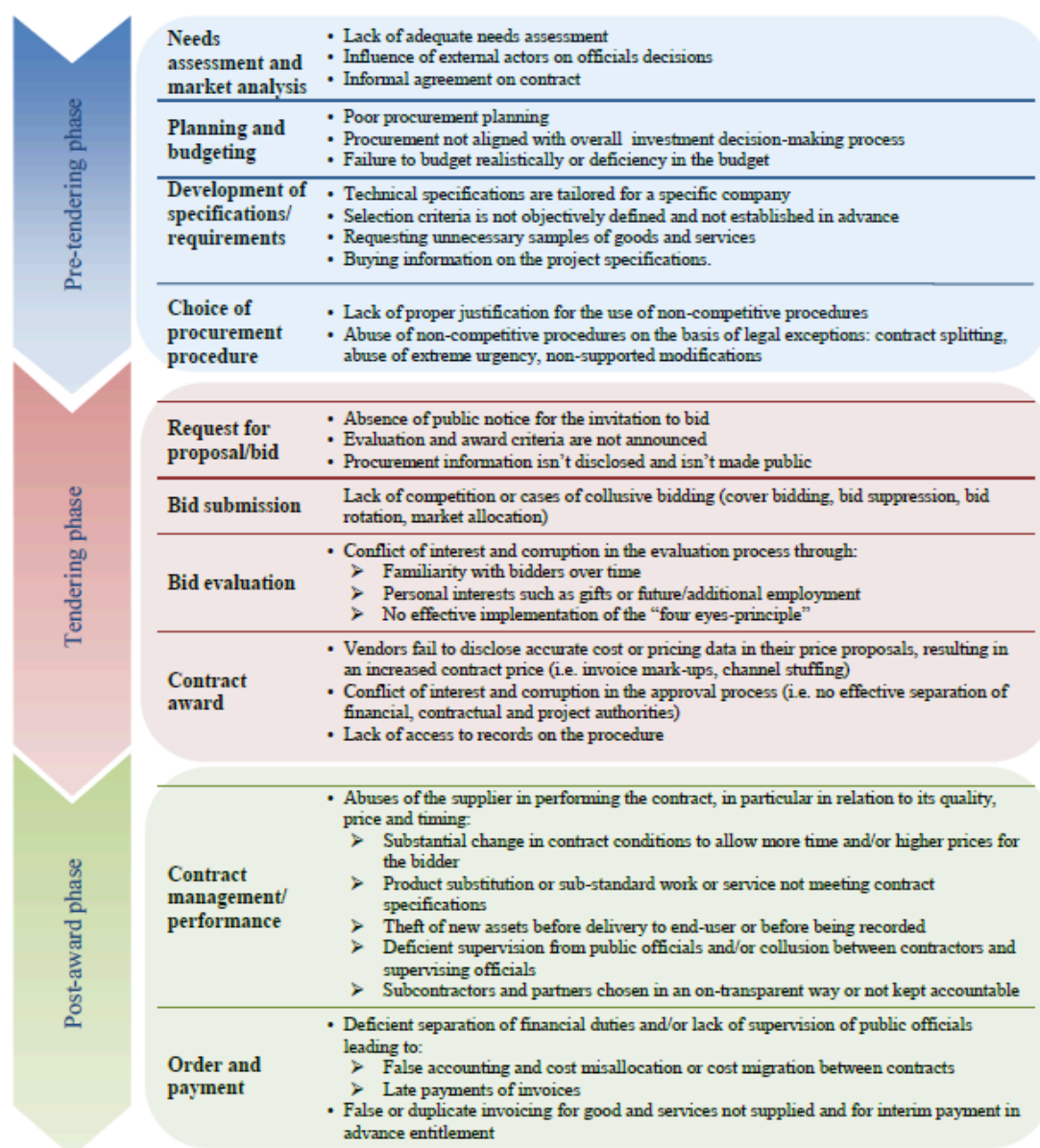


Fig 2

Chapter - 2

2.1 SYSTEM DEVELOPMENT

The Pardarsh project envisions a comprehensive digital platform designed to enhance transparency and accountability in public procurement processes. To achieve its objectives, Pardarsh consists of three key interfaces: for the **Public**, **Government Authorities**, and **Contractors**. These interfaces will be accessible through two platforms— a mobile based application and a web-based application. Each interface will serve a distinct purpose, catering to the specific needs and roles of different stakeholders involved in public procurement. To develop a seamless experience across these interfaces, we are utilizing modern technologies like Flutter, Firebase, React, Node.js, Express.js, and MongoDB.

This chapter discusses the design and development of the system's interfaces, detailing the functionalities, technologies, and workflow for the App and Web Platform.

Interface for Public Users:

The **Public Interface** is a key feature of the Pardarsh platform, empowering citizens to actively participate in monitoring public procurement projects. It provides access to detailed, real-time data about tenders, project costs, contractor progress, and other vital project-related information.

Project Transparency: Public users can search for and view open and awarded tenders, along with detailed cost breakdowns for materials, labor, taxes, and other expenditures. This transparency ensures citizens can track where and how public funds are spent.

Project Updates: Users can monitor the progress of ongoing projects, including the completion of milestones and adherence to timelines. This information is regularly updated by contractors, ensuring it is always up-to-date.

Feedback and Rating System: The platform allows the public to rate contractors based on their performance, timeliness, and the quality of work. Public feedback helps maintain high standards in public procurement.

Complaint Reporting: A robust complaint mechanism enables users to report discrepancies, delays, or issues with projects. This promotes active civic engagement and ensures that issues are addressed by the relevant authorities.

The **Mobile App** for public users is designed using **Flutter** to offer a cross-platform experience, ensuring that both iOS and Android users can seamlessly access the platform. The app is integrated with **Firebase** for real-time data synchronization, notifications, and secure authentication. **Cloud Firestore** is used for storing data, while **Firebase Authentication** manages user sign-ins and profile management.

On the **Web Platform**, the public interface will be developed using **React.js** for a responsive and interactive experience. **Node.js** and **Express.js** will handle the backend functionalities, managing requests and serving data to the front end efficiently. **MongoDB** will be used as the database for storing project data, feedback, and complaints.



Interface for Contractors:

The **Contractor Interface** is designed to allow contractors to efficiently manage their projects and stay compliant with Pardarsh's transparency requirements. It provides tools for contractors to update project progress, upload relevant documents, and engage with both the government and the public.

1. **Project Dashboard:** Contractors have access to a comprehensive dashboard where they can manage project updates, milestones, and financial reports. This feature ensures that contractors stay on track with project requirements and timelines.
2. **Document Upload and Management:** Contractors can upload essential project documentation, including financial reports, purchase orders, and receipts. These documents will be made accessible to the public and government authorities for scrutiny.
3. **Milestone Tracking:** Contractors can track their progress by updating milestones, documenting delays, or outlining challenges. This helps ensure transparency in the procurement process and provides government bodies with a clear view of project status.
4. **Feedback and Rating Management:** Contractors can monitor public feedback and ratings, providing them with insights into their performance. Positive reviews and high ratings will help contractors build a solid reputation for future tenders.

On the **App**, contractors will be able to manage their projects and upload updates easily, thanks to Flutter's cross-platform capabilities. The app will integrate with **Firestore** for data synchronization and **Firestore Authentication** to ensure secure logins and profile management.

For the **Web Platform**, contractors will have access to an intuitive dashboard built using **React.js**. **Node.js** and **Express.js** will handle backend operations like file uploads and project data management, while **MongoDB** will store project details and contractor profiles.

Development Process:

The development of the Pardarsh system involves creating a seamless experience across mobile and web platforms, ensuring that each user interface serves its intended purpose efficiently. The system development follows an iterative approach, starting with basic functionalities and gradually integrating more complex features such as real-time updates, advanced analytics, and automated alerts.

To ensure scalability, security, and performance, the development process focuses on the following:

1. **Real-time Data Syncing:** Using **Firebase** ensures that all platforms stay updated in real time, whether users are accessing the app or web interface.
2. **User Authentication:** **Firebase Authentication** is integrated to manage user profiles and ensure secure logins across all interfaces.
3. **Data Management and Security:** **Node.js**, **Express.js**, and **MongoDB** provide a robust and scalable backend for managing large volumes of project data and documentation.
4. **Feedback Mechanism:** A real-time rating and feedback system is implemented to encourage active participation from the public and ensure contractor accountability.

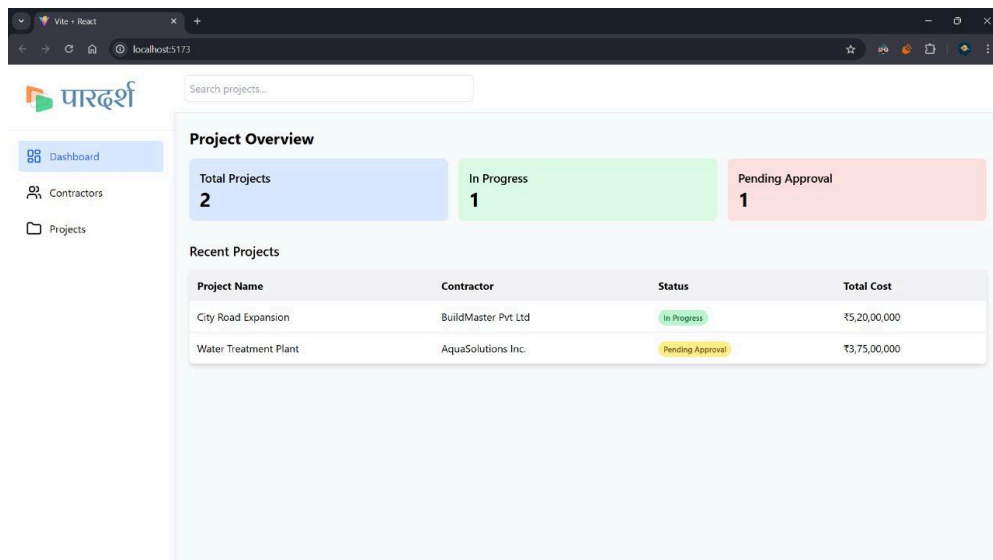


Fig 3

2.2 FUNCTIONAL REQUIREMENTS

The **Pardarsh E-Portal** is designed to enhance transparency in public procurement by enabling real-time project updates and fostering interaction between the public, contractors, and government authorities. The following are the core functional requirements that guide the system's development.

2.2.1. Public Interface

2.2.1.1 Project Transparency

- **Feature:** Public users must have access to detailed information regarding ongoing and completed public procurement projects, including project timelines, costs, and contractor details.
- **Functionality:** Users can search for projects based on criteria such as project type, location, and contractor. All project data, including budget breakdowns, material costs, and progress, must be easily accessible and regularly updated.
- **Real-time Updates:** The system should provide real-time updates on the progress of the project and any changes made to the project details.
- **Data Management:** The backend should allow the system to store, query, and update large volumes of project data effectively, using **Firebase Firestore** and **MongoDB**.

2.2.1.2 Project Progress Tracking

- **Feature:** The platform must provide tracking of project milestones and completion rates.
- **Functionality:** Public users can monitor key milestones, such as the completion of different project phases. Updates will be uploaded by contractors or government authorities.
- **Notification System:** Real-time notifications are required to alert public users when important milestones are completed or if there are any issues with the project (e.g., delays).

2.2.1.3 Feedback and Rating System

- **Feature:** Public users must be able to rate contractors based on their project performance, including timeliness, quality, and adherence to standards.
 - **Functionality:** The rating system must be intuitive, allowing users to leave reviews and scores. This feedback will be visible to contractors and government authorities to ensure accountability.
-

2.2.2. Government Interface

2.2.2.1 Tender Management

- **Feature:** Government authorities should have the ability to post new tenders, close tenders, and view bidding results.
- **Functionality:** The system must allow the government to create, modify, and manage tenders. A list of all open and awarded tenders should be displayed.
- **Data Handling:** The backend must handle multiple tender entries and status updates, ensuring data consistency across platforms.

2.2.2.2 Project Monitoring

- **Feature:** The system must allow government authorities to monitor project progress in real time.
- **Functionality:** Government users should have access to dashboards displaying project milestones, the contractor's work rate, and any discrepancies between expected and actual progress. This helps ensure that contractors adhere to their agreements.
- **Reporting:** A reporting tool should be integrated to allow government authorities to generate reports on project performance, budget utilization, and issues like delays or cost overruns.

2.2.2.3 Public Interaction and Complaint Handling

- **Feature:** The system should include mechanisms for government authorities to interact with the public.
- **Functionality:** Government authorities must be able to respond to public complaints and queries regarding ongoing projects. This interaction should be logged for accountability and transparency.

- **Complaint Resolution System:** A structured complaint management system should allow government bodies to assign complaints to the relevant department and track their resolution.
-

2.2.3. Contractor Interface

2.2.3.1 Project Dashboard

- **Feature:** Contractors must have a comprehensive dashboard to manage their ongoing and completed projects.
- **Functionality:** Contractors should be able to update project milestones, financial reports, and upload necessary documents related to the project.
- **Data Entry and Editing:** Contractors should be able to easily input and edit project data in real time, ensuring accurate reporting of project status.

2.2.3.2 Document Upload and Management

- **Feature:** Contractors must have the ability to upload relevant documentation such as invoices, receipts, and purchase orders.
- **Functionality:** The system should allow contractors to upload documents in various formats (e.g., PDF, DOCX, JPEG) and store them securely.
- **Document Verification:** The platform should have an automated verification process to ensure that uploaded documents meet the required standards before they are made available to the public and government.

2.2.3.3 Milestone Updates

- **Feature:** Contractors need to update progress on key milestones of the project.
 - **Functionality:** Contractors must report on milestones with visual indicators (e.g., percentage completed). These updates will trigger notifications for both the government and the public interface.
 - **Status Reporting:** The system should track milestones and generate notifications based on the contractor's input. Government authorities and the public should be able to access the most up-to-date information about the project's status.
-

2.2.4. Authentication and User Management

2.2.4.1 User Roles and Permissions

- **Feature:** The system should support different user roles with varying levels of access.
- **Functionality:**
 - **Public users** can only view project information and provide feedback.
 - **Government authorities** have access to all project data, tender management, and public interaction tools.
 - **Contractors** can update project progress, upload documents, and track their own projects.
- **Secure Authentication:** The system should use **Firebase Authentication** for secure sign-ins, password recovery, and profile management.

2.2.4.2 Role-based Access Control (RBAC)

- **Feature:** The system must ensure that users can only access data and features they are authorized to.
 - **Functionality:** Admins and government authorities should have full access to project data, while contractors and the public should only have restricted access to specific project information.
-

2.2.5. Notification System

2.2.5.1 Real-time Alerts

- **Feature:** Users (public, government, contractors) should receive real-time notifications for project updates, feedback, and issues.
 - **Functionality:** The notification system should notify users about updates on milestones, contractor ratings, and public complaints.
 - **Technology:** **Firebase Cloud Messaging (FCM)** will be used to send push notifications to mobile users, while email notifications will be sent for web users.
-

2.2.6. Data Security and Privacy

2.2.6.1 Secure Data Handling

- **Feature:** The system should handle all user and project data securely.
- **Functionality:** Data storage should comply with industry standards for data protection. Secure encryption methods must be used for storing sensitive information.
- **Compliance:** The system should comply with relevant data protection regulations (e.g., GDPR) to ensure user data privacy.

2.2.6.2 Backup and Recovery

- **Feature:** The system should implement automated data backups to prevent loss of critical information.
- **Functionality:** Regular backups of project data, tender information, and user data should be stored in secure cloud storage. There should also be a disaster recovery plan in place.

In conclusion, the functional requirements outlined above ensure that the Pardarsh system will be comprehensive, secure, and user-friendly for all stakeholders involved in the public procurement process.

2.3 NON - FUNCTIONAL REQUIREMENTS

2.3.1. Compliance and Regulatory Requirements

2.3.1.1 Government Regulations Compliance

- **Requirement:** The system must comply with all relevant government laws and regulations concerning public procurement, data privacy, and digital transactions.
- **Details:** Compliance with regulations like the **Public Procurement (Preference to Make in India) Order** and **Government e-Marketplace (GeM)** is essential. The platform should also ensure adherence to guidelines for tender management, particularly in sectors related to infrastructure, construction, and government contracts.
- **References:** The system must implement features like e-signatures for authentication and use secure, auditable logging mechanisms in line with Indian e-Governance standards.

2.3.1.2 Contractor Compliance

- **Requirement:** Contractors must adhere to the standards and rules outlined in their contracts with the government.
- **Details:** Contractors must be held accountable for timely submissions of progress updates, budget management, and meeting project milestones. Non-compliance can result in penalties or disqualification from future tenders. The system should incorporate automated checks for compliance (e.g., document verification, milestone verification).
- **Enforcement:** An automated alert system should notify contractors of impending deadlines or issues regarding contract compliance.

2.3.1.3 Data Privacy Compliance

- **Requirement:** The system must comply with data protection laws to ensure the privacy and confidentiality of personal information.
- **Details:** Compliance with **General Data Protection Regulation (GDPR)** for European users, **Data Protection Bill 2019** (India), and other national or international standards related to data privacy should be ensured. This includes securing sensitive information such as financial data, tender information, and user personal details.

- **Security Measures:** Implement **end-to-end encryption**, **role-based access control (RBAC)**, and **data anonymization** for specific personal and financial information.
-

2.3.2. System Performance and Scalability

2.3.2.1 System Availability and Uptime

- **Requirement:** The platform must ensure high availability to support continuous access for users.
- **Details:** The system should aim for an **uptime of 99.9%** or better, ensuring that the platform remains accessible to all users (public, contractors, and government) without significant downtime.
- **Performance Metrics:** Use **load balancing** and **auto-scaling** techniques to handle peak usage, especially during major tenders or project milestones.

2.3.2.2 Scalability

- **Requirement:** The system must be scalable to accommodate increasing numbers of projects, users, and contractors as the platform grows.
 - **Details:** The system architecture should be designed to scale horizontally, adding more resources as demand increases. This includes supporting a growing database of projects, contractors, and public users.
 - **Cloud Infrastructure:** Leverage cloud services (e.g., **AWS**, **Google Cloud**, **Azure**) for **elastic scalability**, ensuring that the system can handle surges in traffic.
-

2.3.3 Performance and Efficiency

2.3.3.1 System Response Time

- **Requirement:** The system should ensure that requests are processed efficiently, with minimal delay.
- **Details:** For typical actions such as querying project data, submitting feedback, and uploading documents, the system should aim for a response time of under **2**

seconds. Critical processes such as milestone updates or government notifications should not exceed **5 seconds**.

2.3.3.2 Database Optimization

- **Requirement:** The backend database (Firestore/MongoDB) must be optimized to handle large volumes of data efficiently.
 - **Details:** Regular database indexing, data partitioning, and query optimization should be implemented to ensure that database access times remain consistent even as the number of projects and users increases. **Caching mechanisms** like **Redis** should be used for frequently accessed data.
-

2.3.4. Security Requirements

2.3.4.1 Secure Authentication

- **Requirement:** The system must implement robust authentication mechanisms to ensure only authorized users can access sensitive information.
- **Details:** Use **multi-factor authentication (MFA)** for government authorities and contractors. The **Firebase Authentication** system will ensure secure user sign-ins, password management, and profile verification. Public users should be able to securely access project data without compromising system security.

2.3.4.2 Role-Based Access Control (RBAC)

- **Requirement:** The system must implement **role-based access control** to ensure that users only access data relevant to their roles.
- **Details:** Government officials, contractors, and public users must have different levels of access to project data, with stringent restrictions on sensitive data such as financial transactions or tender allotment results.

2.3.4.3 Data Encryption

- **Requirement:** All sensitive data must be encrypted during transmission and storage.
- **Details:** **SSL/TLS encryption** will be used for securing data during transmission. Additionally, sensitive data at rest, such as user credentials and financial details, will be encrypted using **AES-256** encryption.

2.3.5. User Experience and Accessibility

2.3.5.1 Intuitive User Interface

- **Requirement:** The system must be easy to use, with an intuitive user interface for all users.
- **Details:** The app and web interfaces should be designed with simplicity and clarity in mind, ensuring that users (public, contractors, government) can easily navigate through the system. The **Flutter** framework for the mobile app will be used to create a responsive, cross-platform interface, while the web platform will be built using **React.js** for smooth user interaction.

2.3.5.2 Mobile and Web Accessibility

- **Requirement:** The platform should be fully accessible on both mobile and web platforms.
- **Details:** The mobile application, built with **Flutter**, should be available on both **Android** and **iOS** devices, with a responsive web version compatible across all major browsers (Chrome, Firefox, Edge). The platform should follow the **WCAG (Web Content Accessibility Guidelines)** to ensure accessibility for users with disabilities.

2.3.6. Integration Requirements

2.3.6.1 Integration with Government Tender Databases

- **Requirement:** The system must integrate seamlessly with government databases for tender allotment and public procurement.
- **Details:** The system should have the ability to access, import, and display tender data from government databases, such as the **GeM (Government e-Marketplace)** or **CPPP (Central Public Procurement Portal)**. The platform should allow government users to upload and sync project details directly with these databases to maintain accuracy and real-time updates.

2.3.6.2 Integration with Contractor Systems

- **Requirement:** The platform must be able to interact with contractor systems for document verification and project status updates.
 - **Details:** Contractors should be able to upload necessary documents directly from their internal project management systems. API integrations with contractor's ERP systems will be used to streamline the process of milestone updates, financial reporting, and documentation verification.
-

2.3.7. Reliability and Fault Tolerance

2.3.7.1 Data Backup and Disaster Recovery

- **Requirement:** The system must implement automatic data backups and a robust disaster recovery process.
- **Details:** Regular data backups should be scheduled to ensure minimal data loss in case of system failure. The system should support **disaster recovery** within 24 hours, ensuring business continuity. The use of cloud-based infrastructure allows for faster data recovery through multiple redundant systems.

2.3.7.2 System Redundancy

- **Requirement:** Redundant systems should be put in place to ensure system reliability.
 - **Details:** **Load balancing** and **failover mechanisms** should be used to ensure that the system continues to operate even if one server goes down. This will minimize downtime and maintain continuous access to the portal.
-

2.3.8 Conclusion

The non-functional requirements outlined above ensure that the **Pardarsh E-Portal** is secure, scalable, compliant with regulations, and capable of delivering high-performance user experiences. By focusing on these requirements, the system will foster trust, ensure operational efficiency, and provide a robust foundation for transparency in public procurement.



REFERENCES

| Research Paper | Link |
|---|---|
| Preventing Corruption in Public Procurement | https://baselgovernance.org/sites/default/files/2020-03/oecd_preventing_corruption_in_public_procurement_2016.pdf |
| National judicial academy report by Rakesh Jain | https://nja.gov.in/Concluded_Programmes/2016-17/P-989_PPTs/5.%20Rakesh%20Jain%20Presentation.pdf |