Submitted by—

1) SOUMYA DAS [2405103470003]

2) RIMI BANERGEE [2405103480003]

3) TANISHA [2405103410009]

4) SOUMYADEEP DAS [2403031460186]

5) SUPRATIM DAS [2403031590065]

6) JAGYANSENI PAIKARAYA [2405103410003]

as

SSIP Gujarat Govt sponsor Hackathon

A logo of a university

AI-generated content may be incorrect.  
**PARUL INSTITUTE OF COMPUTER APPLICATION &**

**PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY**

**CERTIFICATE**

This is to certify that ***Soumya Das, Supratim Das,*** ***Rimi Banergee, Tanisha, Jagyanseni Paikaraya, Soumyadeep Das*** the students of Parul Institute of Computer Application & Parul Institute of Engineering & Technology has/have satisfactorily completed the project entitled “***Soumya tui likhe ne”***

Enrolment No: 2405103470003

Enrolment No: 2403031460186

Enrolment No : 2405103480003

Enrolment No : 2405103410009

Enrolment No: 2405103410003

Enrolment No: 2403031590065

**ACKNOWLEDGEMENT**

The project presented as a part of our academic curriculum was our first experience of this kind. We approached this project not merely as a course requirement but as a meaningful opportunity to learn, explore, and develop skills in commercial software technologies. We are pleased to state that we have successfully achieved our objectives in building and presenting this project.

This accomplishment would not have been possible without the support and encouragement of several individuals.

First and foremost, we express our sincere gratitude to our internal guide and coordinator, and our respected principal, Dr. Priya Swaminarayan, for providing the necessary facilities and guidance throughout the project.

We also take this opportunity to thank our internal guide, Prof. [Sohil Parmar], for his professional insights and support during the industrial development phase.

A heartfelt thanks goes to our parents for their constant encouragement and support, both emotionally and mentally, which motivated us to perform our best.

We are equally grateful to our friends and classmates whose direct and indirect contributions were invaluable and will always be remembered.

This project has given us valuable experience and practical exposure to software development, and it has significantly enriched our academic journey.

[Your Name]

**ABSTRACT**

The Consumer Complaint Tracking System is a process in which a user can register, track, and monitor the status of their complaints to the utility company from any location using the internet. The system provides transparency to consumers by offering real-time status updates from the registration of the complaint to its closure.

The Consumer Complaint Tracking System is a simple project created using PHP, JavaScript, and CSS. The project contains an admin (DISCOM management) side and a user (consumer) side. All management, such as assigning complaints, updating resolution status, and generating reports, is handled from the admin side. For the user section, consumers can register a new complaint, check its current status, and view their complaint history.

This project offers a convenient way for both consumers and the utility company to manage complaints efficiently. The design of this system is simple so that the user won’t encounter any difficulties while working on it. This is a Complaint Tracking system written using PHP/MySQL.

**TABLE OF CONTENT**

The following table: Chapter 1: Introduction,,1 ,1.1 Software Process Implementation,1 ,1.2 System Overview,2 ,1.3 Business Models,2 ,1.4 Project Objectives,2 ,1.5 Project Scope,2 ,, Chapter 2: Requirements and Analysis,,3 ,2.1 Functional & Non-Functional Requirements,3 ,2.2 Software and Hardware Requirements,5 ,2.3 Project Plan & Gantt Chart,6 ,, Chapter 3: Literature Review,,6 ,3.1 Privacy and Security Issues,7 ,3.2 Complaint Tracking Processes,7 ,3.3 Security and Privacy Concerns,10 ,3.4 Waterfall Model,10 ,3.5 SDLC,11 ,3.6 Case Studies,12 Chapter 4: Software & Tool Requirement Analysis,,13 ,4.1 Tools and Techniques,13 ,4.2 Database Details,13 ,4.3 Case Studies of Software Analysis,15 ,, Chapter 5: Software Design,,19 ,5.1 Use Case Diagram,19 ,5.2 System Architecture Diagram,20 ,5.3 Data Flow Diagrams (DFDs),21 ,5.4 Class Diagram,23 ,5.5 Object Diagram,24 ,5.6 Deployment Diagram,25 ,5.7 Use Case Diagrams (Admin/User),26 ,5.8 Sequence Diagrams,28 ,5.9 Activity Diagram,31 ,5.10 Flowchart,32 ,5.11 ER Diagram,33 ,5.12 Collaboration Diagram,34 ,5.13 Component Diagram,35 ,5.14 Association Model,35 ,5.15 State Chart Diagram,36 ,5.16 Communication Diagram,36 Chapter 6: Algorithm and Its Complexity,,37 ,6.1 Algorithms and Pseudocode,37 ,6.2 Complexity Analysis,37 ,6.3 Dry Run and Examples,37 ,6.4 Complaint Search Algorithm & Analysis,38 ,6.5 Login Check Algorithm,39 ,6.6 Graphical Representations,40 ,, Chapter 7: Development Phase,,41 ,7.1 Layout and Form Design,41 ,7.2 Backend & Database Integration,42 ,7.3 Validation and Security Enhancements,46 ,, Chapter 8: Testing,,50 ,8.1 Testing Types and Methods,50 ,"8.2 Performance, Stress & Load Testing",51 ,8.3 Functional Testing & Bug Reports,53 ,8.4 Tools Used & UAT,54 Chapter 9: Conclusion,,56 ,9.1 Project Outcomes and Benefits,56 ,9.2 Future Scope and Impact,57 ,, Chapter 10: Enhancements & Recommendations,,58 References/Bibliography,,59

**CHAPTER 1** **INTRODUCTION**

**1.1 Software Process Implementation – GUVNL Complaint Tracking System** The Complaint Tracking System project follows a structured software development process, which involves key activities such as planning, design, coding, and deployment/maintenance. The project aims to develop a platform for GUVNL consumers to register and track their complaints, including features such as complaint submission forms, a unique complaint ID for tracking, and real-time status updates. The platform supports user registration/login, and the admin panel allows full complaint management by GUVNL staff.

The system includes features such as:

* Complaint registration form
* Real-time status tracking via unique ID
* User registration/login
* Admin dashboard for managing complaints
* Efficient complaint assignment to workforce

**1.2 System Overview** This system refers to the process of digitizing and streamlining the consumer complaint process for DISCOMs. It encompasses technologies like web portals, mobile platforms, and online database management. In the case of GUVNL, the system provides a centralized online platform for consumers to register and track their complaints, facilitating efficient and transparent communication with the utility company.

**1.3 Business Models** The project primarily follows the **B2C (Business-to-Consumer)** model, where GUVNL provides a service directly to its consumers. It also has aspects of a **B2E (Business-to-Employee)** model, as the admin panel is used by GUVNL employees to manage and resolve complaints.

**1.4 Project Objectives**

* **User Registration/Login:** Consumers must register to submit and track complaints.
* **Complaint Submission:** A simple form for users to submit new complaints.
* **Complaint Tracking:** Users can track the status of their complaints using a unique ID.
* **Real-time Updates:** Provides real-time notifications on complaint status changes.

**1.5 Project Scope**

* **Admin Panel:** Full back-end management of complaints, user accounts, and workforce assignment.
* **User Interaction:** Secure front-end for registering complaints and checking status.
* **Complaint Confirmation:** Instant generation of a unique complaint ID and email notification.
* **Educational Use:** The system also serves as a case study for software development courses.

**CHAPTER 2** **REQUIREMENTS AND ANALYSIS**

**2.1 Functional and Non-Functional Requirements** **2.1.1 Problem Definition** The existing GUVNL complaint system is manual and lacks digital integration, leading to inefficiencies and a lack of transparency for consumers.

**Limitations in Existing System:**

* No centralized online platform for complaint registration or tracking.
* Difficulty for consumers to get real-time status updates.
* Manual complaint management causing errors and delays.
* Poor communication between consumers and the utility workforce.

**Proposed System: GUVNL Complaint Tracking System**

* Centralized platform for complaint submission and tracking.
* Easy complaint management and live tracking for both consumers and admins.
* Profile-based system for users and a dashboard for the utility company.
* Real-time status updates via a unique complaint ID.

**2.1.2 Requirements Specification** **2.1.2.1 User Requirements:**

* User registration with name, email, mobile, and password.
* Secure login and password recovery.
* A form to register a new complaint.
* The ability to view a list of previously submitted complaints.
* Real-time status tracking for each complaint.

**2.1.2.2 Functional Requirements:**

* Complaint submission with a unique ID generation.
* The ability for admins to assign complaints to technicians.
* Real-time status updates for complaints.
* Password reset and profile updates.
* Admin control for managing users and complaints.

**2.1.2.3 Non-Functional Requirements:**

* **Availability:** MySQL for fast data access.
* **Reliability:** Secure and reliable system performance.
* **Security:** Encrypted data and HTTPS protection.
* **Maintainability:** Structured data storage for easy updates.
* **Usability:** Easy-to-use interface for both consumers and admin users.

**2.2 Software and Hardware Requirements** **2.2.1 Software Requirements**

* **AptanaStudio/Sublime Text:** Integrated Development Environment (IDE) for coding.
* **XAMPP:** Local server environment including Apache, MySQL, and PHP.
* **PHP:** Server-side scripting language for backend operations.
* **MySQL:** Relational database system for storing user and complaint data.
* **HTML, CSS, JavaScript:** For frontend design and interactivity.
* **Bootstrap:** Front-end framework for responsive design.

**2.2.2 Hardware Requirements**

* **Operating System:** Windows XP / 7 / 8 / 8.1 (32 or 64-bit)
* **Processor:** Minimum 1.5 GHz required.
* **RAM:** 1 GB or more recommended.
* **Hard Disk Space:** At least 2 GB free space.

**2.3 Project Plan & Gantt Chart** The following table: Task Name,Duration,Start,Finish,Predecessors Project topic deciding,3 days,07/02/2025,11/02/2025, Collecting data,8 days,12/02/2025,21/02/2025,1 Problem definition,19 days,24/02/2025,20/03/2025,2 System design,13 days,21/03/2025,08/04/2025,3 Problem evaluation,4 days,09/04/2025,14/04/2025,4 Define function and behavior,11 days,15/04/2025,28/04/2025,5 Requirement analysis,17 days,29/04/2025,21/05/2025,6 Implementation and coding,23 days,22/05/2025,24/06/2025,7 Unit testing,4 days,25/06/2025,30/06/2025,8 Integration and validation,7 days,01/07/2025,09/07/2025,9 System Testing,6 days,10/07/2025,17/07/2025,10 Table 1 Project Plan

**CHAPTER 3** **LITERATURE REVIEWS**

**3.1 Privacy and Security Issues** The primary concern for users submitting complaints online is the privacy and security of their personal information. The system must use strong encryption methods (like SSL) to protect data both during transmission and while stored on the server.

**3.2 Processes in the Complaint Tracking System:**

* **Sign In Process:** A user enters credentials to log in and access their dashboard.
* **Sign Up Process:** A new user provides details to register an account.
* **Complaint Submission Process:** A user fills out a form to submit a new complaint.
* **Tracking Process:** The user enters a unique ID to get the current status of their complaint.
* **Admin Process:** The admin logs in to view and manage all submitted complaints.
* **Resolution Process:** The assigned technician updates the complaint status, and the consumer is notified.

**3.3 Security and Privacy Concerns:** Customers are often wary of privacy and security issues when submitting personal data online. Security measures like SSL encryption are essential, but customers should also be proactive in safeguarding their information.

**3.4 Waterfall Model:** The project development follows a sequential model with distinct phases:

* **Requirement Gathering and analysis:** Capturing all requirements for the system.
* **System Design:** Designing the system architecture and database.
* **Implementation:** Developing the system in small units.
* **Integration and Testing:** Integrating units and testing the entire system.
* **Deployment:** Releasing the system for use by GUVNL.
* **Maintenance:** Fixing issues and releasing updates.

**3.5 System Development Life Cycle (SDLC):** The project followed the SDLC methodology to ensure a structured and well-documented development process.

* **Planning:** Defining project scope, timeline, and deliverables.
* **Data Requirement & Analysis:** Gathering data to meet business needs and user expectations.
* **Design:** Designing the system's architecture, including diagrams.
* **Development:** Coding the system using PHP, MySQL, HTML, CSS, and JavaScript.
* **Integrate & Testing:** Testing the system for bugs, performance, and functionality.
* **Implementation:** Deploying the final system to the production environment.

**3.6 Case Studies**

* **Case Study 1:** Addresses the need for a real-time status tracker for users, a feature missing in the current manual system.
* **Case Study 2:** Details how the Admin Panel was designed to allow DISCOM staff to efficiently assign complaints and monitor resolution times.

**CHAPTER 4** **SOFTWARE & TOOL REQUIREMENT ANALYSIS**

**4.1 Tools and Techniques** The development of this project utilized a variety of tools and technologies:

* **PHP:** For server-side scripting and backend logic.
* **XAMPP:** For a local development environment.
* **MySQL Workbench:** For database design and management.
* **HTML, CSS, JavaScript:** For frontend development.
* **Bootstrap:** For a responsive and mobile-first user interface.
* **Sublime Text:** A powerful code editor.
* **GitHub:** For version control and collaborative development.

**4.3 CASE STUDIES OF SOFTWARE ANALYSIS**

* **Case Study: Consumer Complaint Process**
  + A consumer visits the portal and registers a complaint.
  + A unique complaint ID is generated and sent to the consumer.
  + The consumer can use the ID to track the status of their complaint.
  + The complaint is routed to the appropriate DISCOM team.
* **Case Study: Admin Panel**
  + The admin oversees the entire complaint management lifecycle.
  + The admin can view new complaints, assign them to technicians, and update the status.
  + The admin can also manage user accounts and generate reports.

**CHAPTER 5** **SOFTWARE DESIGN**

**5.1 Use Case Diagram** Illustrates the interactions between consumers, admins, and the system.

* **Actors:** Consumer, Admin.
* **Use Cases:** Register Complaint, Track Status, Manage Complaints, Assign Technician, Update Status.

**5.2 System Architecture Diagram** Shows the interaction between the user's browser, the web server (Apache), the application server (PHP), and the database server (MySQL).

**5.3 DATA FLOW DIAGRAM (DFD)**

* **‘0’ level DFD:** Shows the overall flow of data between the Consumer, Admin, and the GUVNL Complaint System.
* **‘1’ level DFD:** Provides more detail on the data flow within the Admin and Consumer panels.

**5.4 CLASS DIAGRAM** The class diagram represents the static structure of the system with classes for User, Complaint, Admin, Technician, and StatusHistory.

**5.5 OBJECT DIAGRAM** Shows a snapshot of the objects and their relationships at a specific point in time.

**5.6 DEPLOYMENT DIAGRAM** Models the physical deployment of the software on the server and the interaction with the user's device via the internet.

**5.7 USE CASE DIAGRAMS (Admin/User)**

* **Consumer Use Case:** Shows that a user can register, log in, submit a complaint, and check its status.
* **Admin Use Case:** Shows the admin can manage users, complaints, and technicians.

**5.8 Sequence Diagrams**

* **Non-Register User Sequence Diagram:** Shows the steps for a new user to sign up, submit a complaint, and receive a tracking ID.
* **Registered User Sequence Diagram:** Details the process for a logged-in user to submit a complaint.
* **Admin Sequence Diagram:** Illustrates the process of an admin logging in, viewing a new complaint, and assigning it.

**5.9 Activity Diagram** Represents the workflow of a consumer submitting and tracking a complaint, showing the different states and decision points.

**5.10 FLOWCHART** A simple flowchart showing the steps a user takes from logging in to submitting a complaint.

**5.11 ER DIAGRAM** A graphical representation of the entities in the system (User, Complaint, Technician) and the relationships between them.

**5.12 COLLABORATION DIAGRAM** Shows the interaction and relationships between the objects in the system during a scenario, like complaint submission.

**5.13 COMPONENT DIAGRAM** Illustrates the architecture of the software components (Consumer Portal, Admin Dashboard, Database Module) and the dependencies between them.

**5.14 ASSOCIATION MODEL** Shows the relationship between the main processes: Login, Complaint Submission, Tracking, and Resolution.

**5.15 STATE CHART DIAGRAM** Shows the different states of a complaint (e.g., New, Assigned, In Progress, Resolved) and how it transitions between states.

**5.16 COMMUNICATION DIAGRAM** Similar to the sequence diagram, it focuses on the collaboration of objects during a process.

**CHAPTER 6** **ALGORITHM AND ITS COMPLEXITY**

**6.1 Algorithm and Pseudocode** Algorithms are used for key operations like logging in, searching for a complaint, and assigning a technician. Pseudocode provides a structured, language-agnostic representation of this logic.

**6.4 Complaint Search Algorithm** This algorithm is used to find a specific complaint based on its unique ID.

* **Algorithm:** A linear search algorithm to scan the complaints database for a matching ID. A binary search could be used if the data is sorted.
* **Complexity Analysis:**
  + **Best Case:** O(1) - The complaint is found on the first attempt.
  + **Worst Case:** O(n) - The algorithm must check every complaint in the database to find the match or confirm it doesn't exist.

**6.5 Login Check Algorithm** This is a critical security algorithm for both consumers and admins.

* **Algorithm Steps:**
  1. Start
  2. Get username and password from the login form.
  3. Hash the input password.
  4. Query the database to find the user with the given username.
  5. Compare the hashed password with the stored hashed password.
  6. If matched, start a session and redirect to the user or admin dashboard.
  7. If not matched, return an error message.
  8. Exit

**CHAPTER 7** **DEVELOPMENT PHASE**

**7.1 Development Phase – 1: Layout / Form Designing** This phase focused on designing the user interface using HTML5, CSS3, and Bootstrap. Wireframes were created for the main pages, including the homepage, user registration, complaint submission form, and the admin dashboard. The design was made responsive to ensure it works on desktops, tablets, and mobile devices.

**7.2 Development Phase – 2: Backend & Database Integration** The static design was integrated with a dynamic backend using PHP and a MySQL database.

* **Database Design:** A relational database named guvnl\_complaints was designed with tables to handle users, complaints, and status history.
* **Table Schemas:**
  + users (user\_id, email, password, role)
  + complaints (complaint\_id, user\_id, subject, description, status)
  + status\_history (history\_id, complaint\_id, technician\_id, status\_update)
* **Backend Logic:** PHP scripts were written to handle form submissions, database queries, and session management.

**7.3 Development Phase – 3: Validation and Enhanced Security** The system was enhanced with comprehensive validation and security measures.

* **Client-side Validation:** JavaScript was used to validate fields like email format, mobile number length, and password strength in real-time.
* **Server-side Validation:** PHP was used to sanitize inputs, prevent duplicate email registrations, and check file uploads.
* **Security:** Passwords are securely hashed using password\_hash() to protect them from breaches. All database queries use prepared statements to prevent SQL injection attacks.

**CHAPTER 8** **TESTING**

The testing phase was crucial to ensure the system's quality and stability.

* **Types of Testing:** Functional, Performance, Stress, and System testing were all conducted.
* **Functional Testing:** Ensured all features, such as complaint submission and status tracking, worked as expected.
* **Performance and Load Testing:** Tested the system's responsiveness and stability under various user loads. The system was found to be stable under expected traffic.
* **Bug Reports:** Bugs identified during testing (e.g., error messages not displaying, UI overlaps) were logged and resolved.
* **User Acceptance Testing (UAT):** A prototype was reviewed by sample users who found it easy to use and navigate, validating the system’s effectiveness.

**CHAPTER 9** **CONCLUSION**

This project successfully developed a modern, user-friendly, and transparent online platform for GUVNL consumer complaints. It addresses the key issues of the existing manual system by providing real-time tracking and efficient management tools for the utility company. The system not only enhances consumer satisfaction but also improves the overall efficiency of the power system by streamlining the complaint resolution process.

**CHAPTER 10** **ENHANCEMENTS & RECOMMENDATIONS**

The project has been successfully implemented, and while it meets all core requirements, several enhancements can be made to improve it further.

* **Mobile App Integration:** Develop native Android and iOS apps for consumers and technicians to access the system on the go.
* **SMS/Email Notifications:** Implement automatic notifications to users for status updates on their complaints.
* **GIS Integration:** Integrate with a Geographic Information System (GIS) to visually map fault locations for technicians.
* **Analytics and Reporting:** Add advanced reporting features for DISCOM management to analyze common fault types and resolution times.
* **Feedback System:** Allow consumers to provide feedback and rate the service after a complaint has been resolved.

**REFERENCES/BIBLIOGRAPHY**

* Google Search Engine for general information
* Web Technologies Black Book
* Keeves – PHP & MySQL
* Beginning PHP5 – Wrox Press
* www.w3schools.org – Online Tutorials
* Murach’s PHP/MySQL
* Additional Academic Sources:
  + Arnold, M. J. et al. (2005). Customer delight in retail. Journal of Business Research.
  + Martin, B. A. S. (2012). Interpersonal touch effects. Journal of Consumer Research.
  + Harada, M. et al. (1996). Personal ID by face images. IEICE Report.
  + Goldberg, L. R. (1993). Personality traits. American Psychologist.
  + John, O. P. et al. (2008). Big-Five Trait Theory. Guilford Press.
  + Phillips, J. J. et al. (2001). Human Resource ROI. ASTD Publications.
  + Kaner, C. (2006). Exploratory Testing. QAI Conference.
  + Kitchenham, B. et al. (2002). Software Metrics. IEEE Xplore.
  + Runeson, P., & Höst, M. (2009). Case Studies in Software Engineering. Springer.
  + Stanford Online Courses – online.stanford.edu