INTRODUCTION

1. INTRODUCTION

Ulliyeri-e-Panchayat embodies a visionary approach towards modernizing rural governance, placing the needs and aspirations of local communities at its core. By leveraging state-of-the-art technology, the portal endeavors to bridge the digital divide, enhance communication channels, and promote transparent decision-making within the Panchayat system. Through the seamless integration of digital platforms, Ulliyeri-e-Panchayat portal aims to democratize access to essential services, ensuring that every resident has the means to engage with their local governance structure effortlessly. By developing a user-friendly platform that connects residents with local service providers, the portal not only streamlines access to essential amenities but also stimulates local employment opportunities and fosters economic growth.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, problems and using the information to recommend improvements of the system.

2.1 Existing system

Generally, to get a service from the panchayat, the user has to go there. As many people come to the panchayat, the user has to wait there to get the service. Sometimes many benefits from the panchayat are not known to the deserving people. For maintaining records lots of paper gets wasted. Maintaining all the details like ward member, management, user is very difficult job.

2.2 Proposed system

The existing system will be replaced by modern computer technology, this system ease the work of the society on the daily basis. It will reduce time consumption of task to be performed. All the work will be generated on one click only. User can view the services provided by panchayat and apply very easily. The user himself is able to quickly connect to the workers within the panchayat.

2.3 Module Description

This project has five modules

Admin (Panchayat President):

- Manage panchayat secretary
- Add category for staff
- Manage staff
- Add ward
- Manage ward members
- Manage users
- Manage meeting
- Manage notification
- Manage panchayat news
- View history
- View complaint and reply

Secretary:

- View services
- View application form from users
- manage application form
- send reason for rejected applications
- view meeting
- view notification
- view news

Staff:

- Add services
- Add application form for services

- View application form from users
- Manage application form
- Send reasons for rejected application
- View meeting
- View notification
- View news

Ward member:

- Manage meeting
- View meeting from panchayat president
- Manage notification
- View notification from panchayat president
- View complaint and reply
- View news

Users:

- Register
- Login
- View services and details
- View application form and apply
- View application form status
- View reason for rejected application
- Post complaint to president, ward member
- View reply
- View meeting from ward member
- View notification
- View news

2.4 Sprint

Sprint 1

Module	**Task**	**Hours for completion**	**Expected date of completion**	**Actual date of completion**	**Reason for deviation**
President	login	1hr	29/10/24	29/01/24	
	Manage secretary	2hr	29/10/24	29/10/24	
	Add category for staff	1hr	29/10/24	29/10/24	
	Manage staff	2hr	29/10/24	29/10/24	
	Add ward	1hr	30/01/24	30/01/24	
	Manage ward members	2hr	30/01/24	30/01/24	
	Manage meeting	1hr	30/01/24	30/01/24	
	Manage notification	1hr	31/01/24	31/01/24	
	Mange news	1hr	31/01/24	31/01/24	
	View history		31/01/24	31/01/24	
	View complaint	1hr			
	validation				

Sprint 2

Module	**Task**	**Hours for completion**	**Expected date of completion**	**Actual date of completion**	**Reason for deviation**
Secretary	Login	1hr	01/02/24	01/02/24	
	View services	2hr	01/02/24	01/02/24	
	Manage application verified by staff	2hr	01/02/24	01/02/24	
	View meeting	1hr	02/02/24	02/02/24	
	View notification	1hr	02/02/24	02/02/24	
	View news	1hr	02/02/24	02/02/24	
Staff	Login	1hr	03/02/24	02/02/24	
	Manage services	2hr	02/02/24	02/02/24	
	Add application form	2hr	02/02/24	02/02/24	
	Manage application form from user	4hr	03/02/24	03/02/24	
	View meeting	1hr	03/02/24	03/02/24	
	View notification	1hr	04/02/24	04/02/24	
	View news	1hr	04/02/24	04/02/24	
	validation	2hr	05/02/24	05/02/24	

Sprint 3

Module	**Task**	**Hours for completion**	**Expected date of completion**	**Actual date of completion**	**Reason for deviation**
Ward member	Login	1hr	06/02/24	06/02/24	
	Manage meeting	2hr	07/02/24	07/02/24	
	View meeting from president	1hr	08/02/24	08/02/24	
	Manage notification	2hr	09/02/24	09/02/24	
	View notification from president	1hr	10/02/24	10/02/24	
	Manage complaint	2hr	12/02/24	12/02/24	
	View news	1hr	14/02/24	14/02/24	
User	Register	2hr			
	Login	1hr			
	View services	2hr			
	Apply for a service	3hr			
	View application status	3hr			

	Post complaint	1hr			
	View reply				
	View meetings	1hr	20/03/24	20/03/24	
	View notifications	1hr	21/03/14	21/03/14	
	View news				
	validation				

2.5 User stories

Admin (Panchayat President):

- As a president I want to be able to efficiently manage secretary so that I can add, view, update the details of secretary.
- As a president I want to be able to efficiently manage staff so that I can add, view, update the details of staff.
- As a president I want to be able to efficiently manage ward members so that I can add, view, update the details of ward members.
- As a president I want to be able to efficiently manage users so that I can accept or reject them.
- As a president I want to be able to manage meeting efficiently, scheduling and organizing.
- As a president I want to be able to view complaint and reply to complaint
- As a president I want to be able to manage notifications.
- As a president I want to be able to manage news.

Secretary:

- As a secretary I want to be able to view services
- As a secretary I want to be able to view application form verified by staff
- As a secretary I want to be able to accept or reject application forms
- As a secretary I want to be able to send reason for rejected application forms
- As a secretary I want to be able to view meetings
- As a secretary I want to be able to view notification
- As a secretary I want to be able to view news

Staff:

- As a staff I want to be able to view notification
- As a staff I want to be able to add services
- As a staff I want to be able to add application forms

- As a staff I want to be able to view meeting
- As a staff I want to be able to view application form and accept or reject
- As a staff I want to be able to send reason for rejected application forms
- As a staff I want to be able to view news

Ward member:

- As a ward member I want to be able to manage meeting
- As a ward member I want to be able to view meeting from president
- As a ward member I want to be able to manage notification
- As a ward member I want to be able to view notification from president
- As a ward member I want to be able to view complaint and reply
- As a ward member I want to be able to view news

User:

- As a user I want to be able to view services
- As a user I want to be able to view application form and apply
- As a user I want to be able to view application form status
- As a user I want to be able to view reason for rejected applications
- As a user I want to be able to view post complaint and view reply
- As a user I want to be able to view services view meeting
- As a user I want to be able to view services view notification
- As anuser I want to be able to view news

FEASIBILITY STUDY

3. FEASIBILITY STUDY

An analysis of the ability to complete a project successfully, taking intoaccount legal, economic, technological, scheduling, and other factors is considered a feasibility study. Rather than just diving into a project and hoping for the best, feasibility study allows project managers to investigate the possible negative and positive outcomes of a project before investing too much money and time.

3.1 Economic feasibility

The economic analysis is done to determine the benefits and savings that are expected from the candidate system and compare them with costs. Thus, coming to a conclusion on whether the system is economically feasible or not. This system is cost effective as well as time effective, thereby making it economically feasible. This study presents tangible and intangible benefits from the project by comparing the developments and operational costs. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility.

3.2 Technical feasibility

The technical requirements for the system are economic and it does not use additional software. This application is developed using MERN Stack, whose development kit are easily available and free of cost, thus making our system technically feasible.

3.3 Operational feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. The new proposed system is very much useful to the users and there for it will accept broad audience.

3.4 Behavioral feasibility

This analysis involves how it will work when it is installed and the assessment of the political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept a broad audience.

3.5 Software feasibility

Even though this application is developed in a very high software environment, it is also supported by many other environments with minimal changes. The system is fully feasible to be executed on any kind of operating systems and browsers.

3.6 Hardware feasibility

Software can be developed with the existing resources. But the existing resources may or may not be used to produce hardware. If no hardware is newly bought for a project, then software is said to achieve hardware feasibility. The system is hardware-wise feasible because it needs absolutely no new hardware.

SOFTWARE ENGINEERING PARADIGM

4. SOFTWARE ENGINEERING PARADIGM

The software engineering paradigm which is also referred to as a software process model or Software Development Life Cycle (SDLC) model is thedevelopment strategy that encompasses the process, methods and tools. SDLC describes the period of time that starts with the software system being conceptualized.

4.1 Agile model

Agile SDLC model is a combination of iterative and incremental processmodels with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. At the end of the iteration, a working product is displayed to the customer and important stakeholders. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks.

At the end of the iteration, a working product is displayed to the customer and important stakeholders. Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it. When user approach software development in a particular manner, it's generally good to live by these values and principles and use them to help figure out the right things to do given users particular context. One thing that separates Agile from other approaches to software development is the focus on the people doing the work and how they work together. Solutions evolve through collaboration between self-organizing cross-functional teams utilizing the appropriate practices for their context.

4.2 Scrum

Scrum is an agile framework for managing knowledge work, with an emphasis on software development. It is designed for teams of three to nine members, who break their work into actions that can be completed within time boxed iterations, called "sprints", no longer than one month and most commonly two weeks, then trackprogress and re-plan in 15-minute stand-up meetings, called daily scrums.

Scrum is an iterative and incremental framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the" traditional, sequential approach to product development, and enables teams to self organized by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines involved.

Scrum is a framework that helps teams work together. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.

While the scrum is talking about is most frequently used by software development teams, its principles and lessons can be applied to all kinds of teamwork. This is one of the reasons scrum is so popular. Often thought of as an agile project management framework, scrum describes a set of meetings, tools, and roles that work in concert to help teams structure and manage their work.

Scrum is the most common agile framework, and the one most people start with. Agile practices on the other hand, are techniques applied during phases of the software development lifecycle. Planning poker for example, is a collaborative estimation practice designed to encourage team members to share their understanding of what done means. The process is quite fun, and has proven to help foster teamworkand better estimates. Continuous integration (also known as ci) is a common agile engineering practice where code changes are integrated into the main branch frequently. An automated build verifies changes, leading to a reduction in integration debt and a continually shippable main branch. These practices,

like all agile practices, carry the agile label, because they are consistent with the principles in the agilemanifesto.

In the project management, scrum, sometimes written scrum or scrum, is a framework for developing, delivering, and sustaining products in a complex environment, with an initial emphasis on software development, although it has been used in other fields including research, sales, marketing and advanced technologies. It is designed for teams of ten or fewer members, who break their work into goals that can be completed within time-boxed iterations, called sprints, no longer than one month and most commonly two weeks. The scrum team assess progress in time-boxed daily meetings of 15 minutes or less, called daily scrums (a form of stand-up meeting). At the end of the sprint, the team holds two further meetings: the sprint review which demonstrates the work done to stakeholders to elicit feedback,and sprint retrospective which enables the team to reflect and improve.

A key principle of scrum is the dual recognition that customers will changetheir minds about what they want or need and that there will be unpredictable challenges-for which a predictive or planned approach is not suited. As such, scrum adopts an evidence based empirical approach accepting that the problem cannot be fully understood or defined up front, and instead focusing on how to maximize the team's ability to deliver quickly, to respond to emerging requirements, and to adapt to evolving technologies and changes in market conditions. Many of the terms used in scrum (e.g., scrum master) are typically written with leading capitals (e.g., scrum master) or as conjoint words written in camel case (e.g., scrum master). To maintain an encyclopedic tone, however, this article uses normal sentence case for these terms- unless they are recognized marks. This is occasionally seen written in all -capitals, as scrum. The word is not an acronym, so this is not correct; however, it likely arose due to an early paper by ken schwaber which capitalized scrum in its title. While the trademark on the term scrum itself has been allowed to lapse, so that it is deemed as owned by the wider community rather than an individual, the leading capital is retained-except when used with other words.

SYSTEM REQUIREMENT SPECIFICATION

5. SYSTEM REQUIREMENT SPECIFICATION

5.1 Software Requirements

One of the most difficult tasks is selecting software, once the system requirement is find out then we have to determine whether a particular software package fits for those system requirements. This section summarizes the application requirement.

- Operating system : Windows 7 or above
- Frond End : Html, CSS, JavaScript
- Back End : Node JS, Express JS
- IDE : Visual Studio
- Database : Mongo DB

5.2 Hardware Requirements

The selection of hardware is very important in the existence and proper working of any of the software. When selecting hardware, the size and capacity requirements are also important. The hardware must suit all application developments**.**

- Processor : Intel core i3 or above
- RAM : 8GB
- SSD : 512 GB

SYSTEM DESIGN

6. SYSTEM DESIGN

System design is the first in the development phase for many engineered products or systems. It may define the process of applying various techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realisation.

6.1 MongoDB

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. The term database design can be used to describe many different parts of the design of an overalldatabase system.

Non-relational model databases, also known as NoSQL databases, are a type of database management system that diverge from the traditional relational model. Instead of relying on tables with predefined schemas and fixed relationships, NoSQL databases use flexible and dynamic data models, such as document-based, key-value, graph, or column-family.

6.2 Collections

In MongoDB, a collection is a grouping of MongoDB documents. It is the equivalent of a table in relational databases. Collections exist within databases and can store multiple documents in a structured format. Each document within a collection can have a unique structure, meaning they don't have to follow a rigid schema like in traditional relational databases.

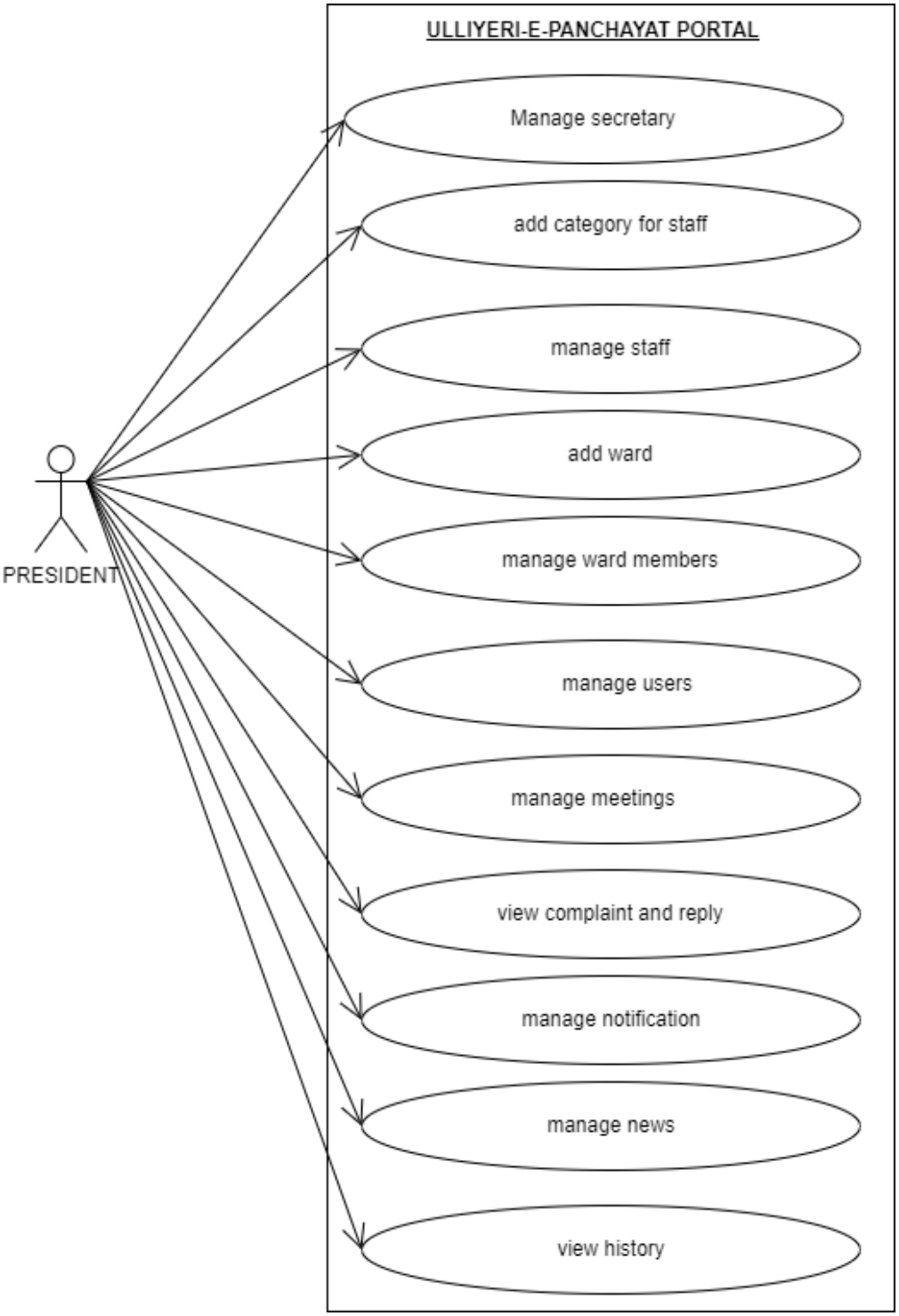
Project Collections

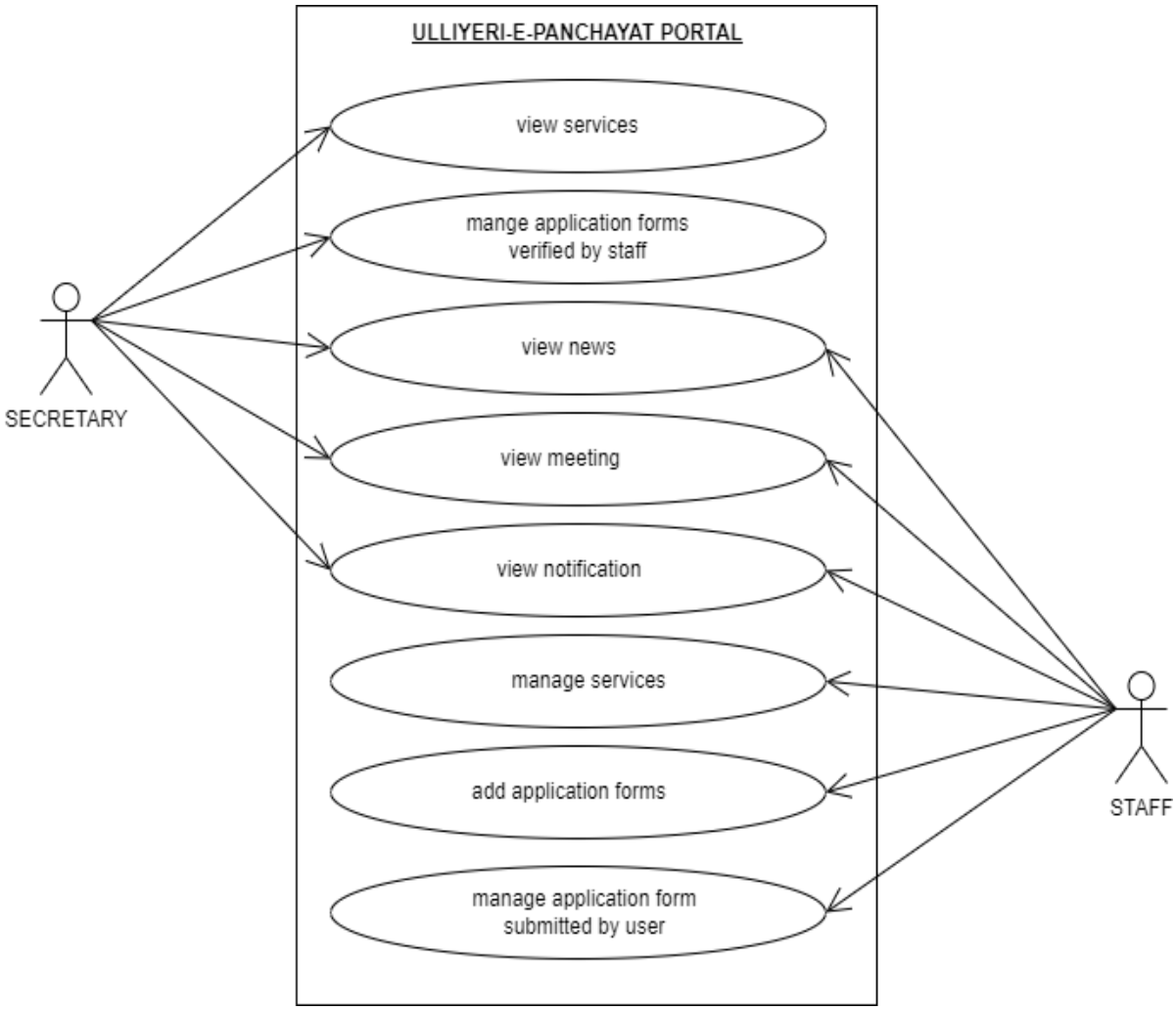
- Users
- Services
- Meetings
- Notifications
- Complaint
- Fields
- Application
- Categories
- Wards
- Complaint

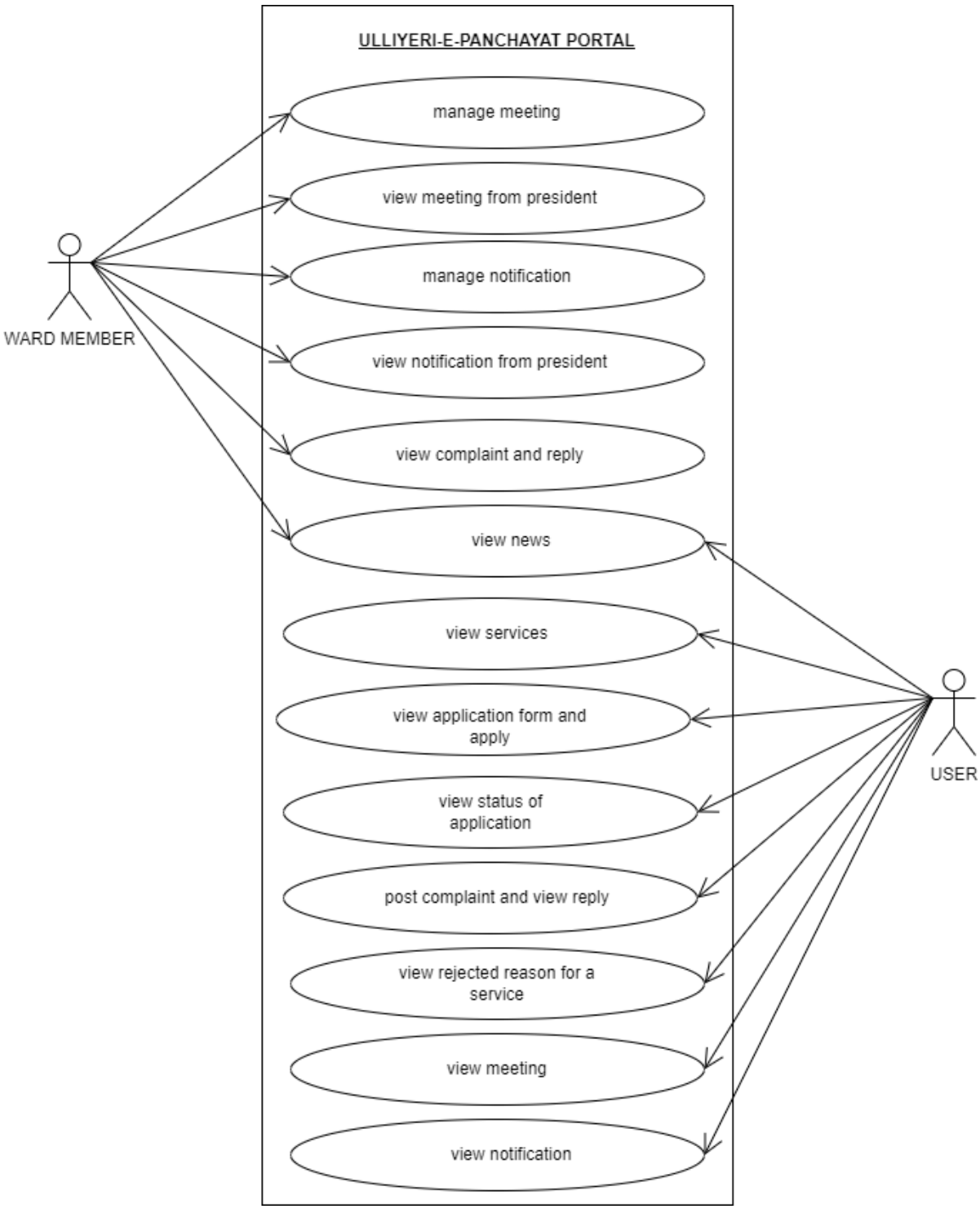
6.3 UML Design

The Unified Modelling Language (UML) is indeed a standardized language used for specifying, visualizing, constructing, and documenting software systems, as well as for business modelling and other non-software systems. It encompasses a collection of best engineering practices that have been proven successful in modelling large and complex systems. UML provides a set of graphical notations that allow software developers and other stakeholders to express and communicate the design of software projects effectively. By using UML, project teams can visualize and explore potential designs, communicate design decisions, and validate the architectural designof the software system. UML diagrams serve as a means to represent various aspects of the system being developed. These diagrams can be used to depict the structure of the system, its behaviour , interactions between components, and the overall flow of activities. The graphical nature of UML diagrams makes them intuitive and easier to understand for both technical and non-technical stakeholders involved in the software development process. UML provides a standardized and widely accepted notation, which promotes consistency and clarity in design documentation. This allows for better collaboration among team members and facilitates the understanding and maintenance of software systems over time. The use of UML in software development can enhance communication, facilitate design exploration, and provide asolid foundation for developing and documenting complex software systems.

6.4 Use Case Diagram




ULLIYERI-E-PANCHAYAT PORTAL
view services
mange application forms verified by staff
view news
view meeting
view notification
manage services
add application forms
manage application form submitted by user
SECRETARY
STAFF


ULLIYERI-E-PANCHAYAT PORTAL
manage meeting
view meeting from president
manage notification
view notification from president
view complaint and reply
view news
view services
view application form and apply
view status of application
post complaint and view reply
view rejected reason for a service
view meeting
view notification
WARD MEMBER
USER

6.5 Scenario

Admin (Panchayat President):

- Admin can login
- Admin can manage panchayat secretary
- Admin can add category for staff
- Admin can manage staff
- Admin can add ward
- Admin can manage ward members
- Admin can manage meeting
- Admin can manage notification
- Admin can manage panchayat news
- Admin can view history
- Admin can view complaint and reply

Secretary:

- Secretary can login
- Secretary can View services
- Secretary can View application form from users
- Secretary can manage application form
- Secretary can send reason for rejected applications
- Secretary can view meeting
- Secretary can view notification
- Secretary can view news

Staff:

- Staff can login
- Staff can Add services
- Staff can Add application form for services

- Staff can View application form from users
- Staff can Manage application form
- Staff can Send reasons for rejected application
- Staff can View meeting
- Staff can View notification
- Staff can View news

Ward member:

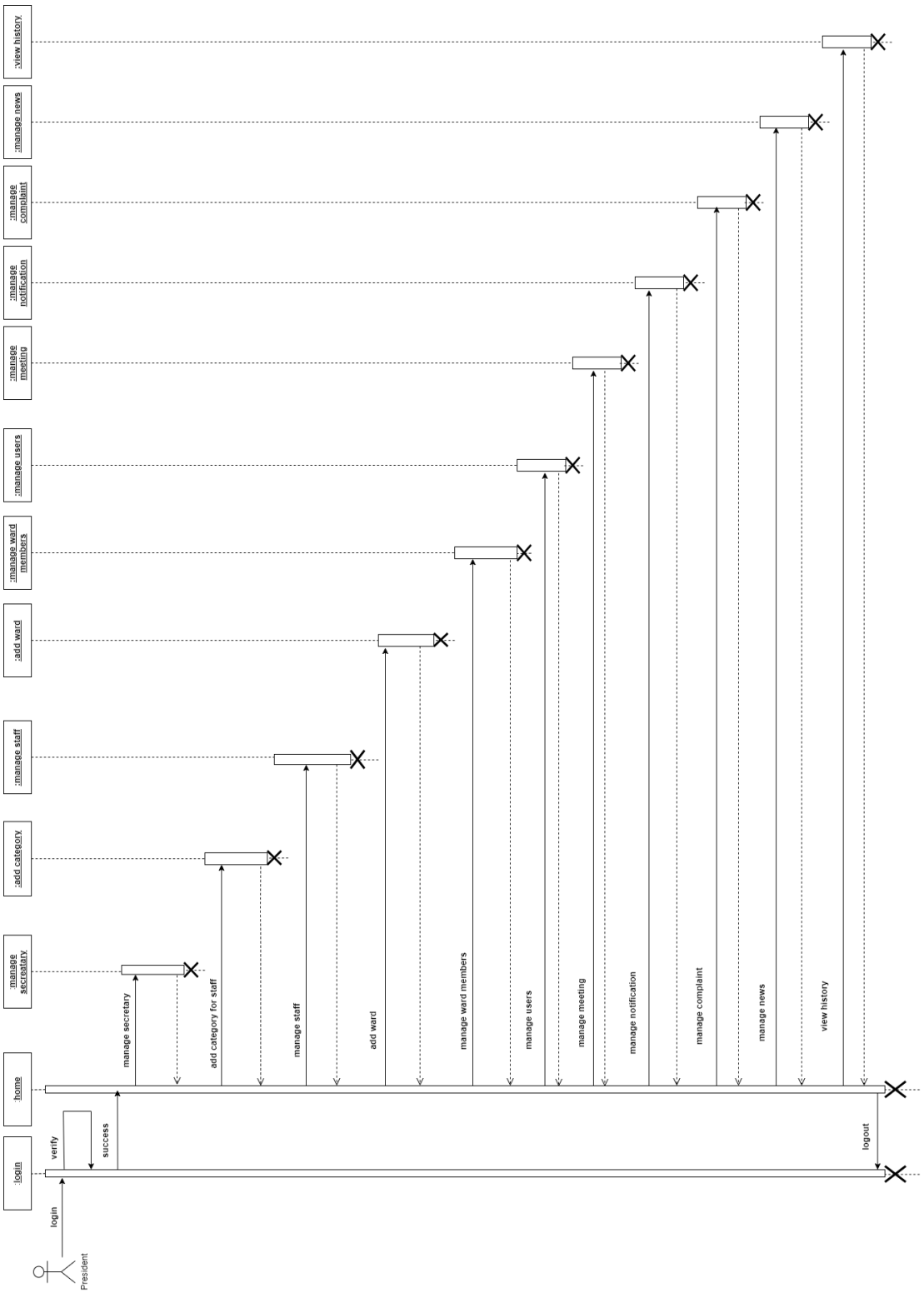
- Ward member can Login
- Ward member can Manage meeting
- Ward member can View meeting from panchayat president
- Ward member can Manage notification
- Ward member can View notification from panchayat president
- Ward member can View complaint and reply
- Ward member can View news

Users:

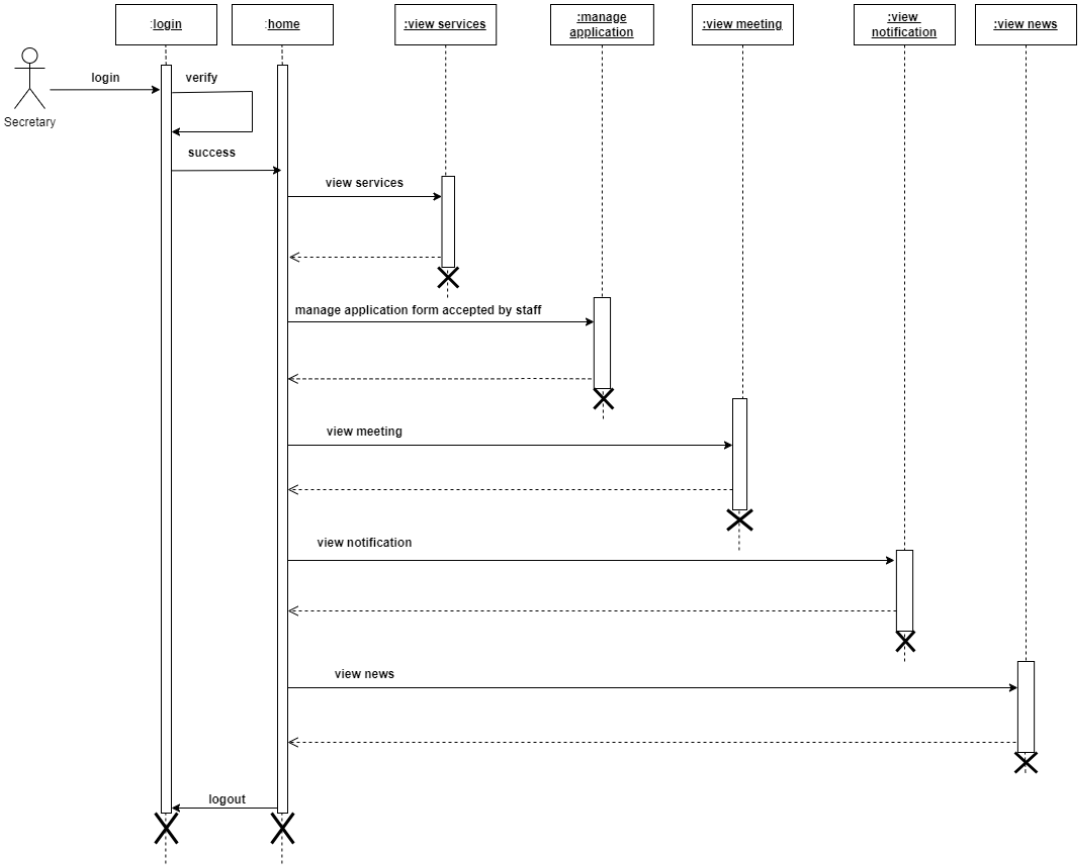
- User can Register
- User can Login
- User can View services and details
- User can View application form and apply
- User can View application form status
- User can View reason for rejected application
- User can Post complaint to president, ward member
- User can View reply
- User can View meeting from ward member
- User can View notification from ward member
- User can View news

6.6 Sequence Diagram

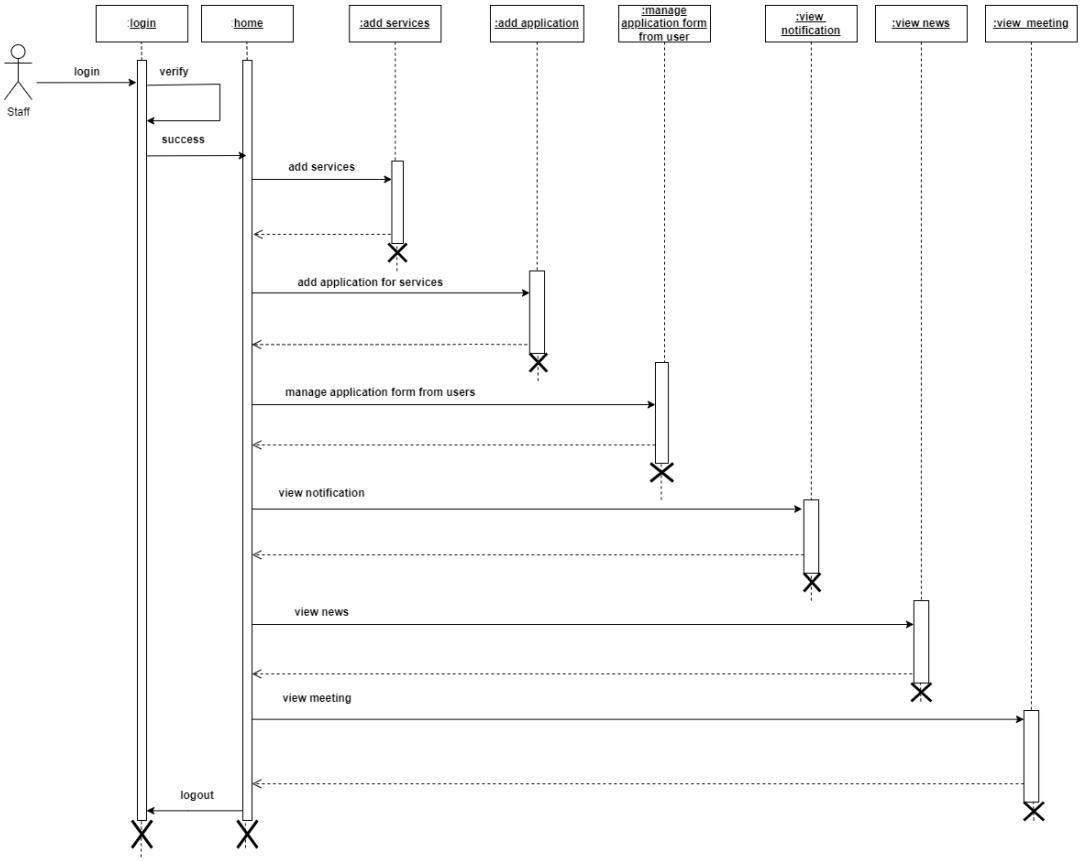
President



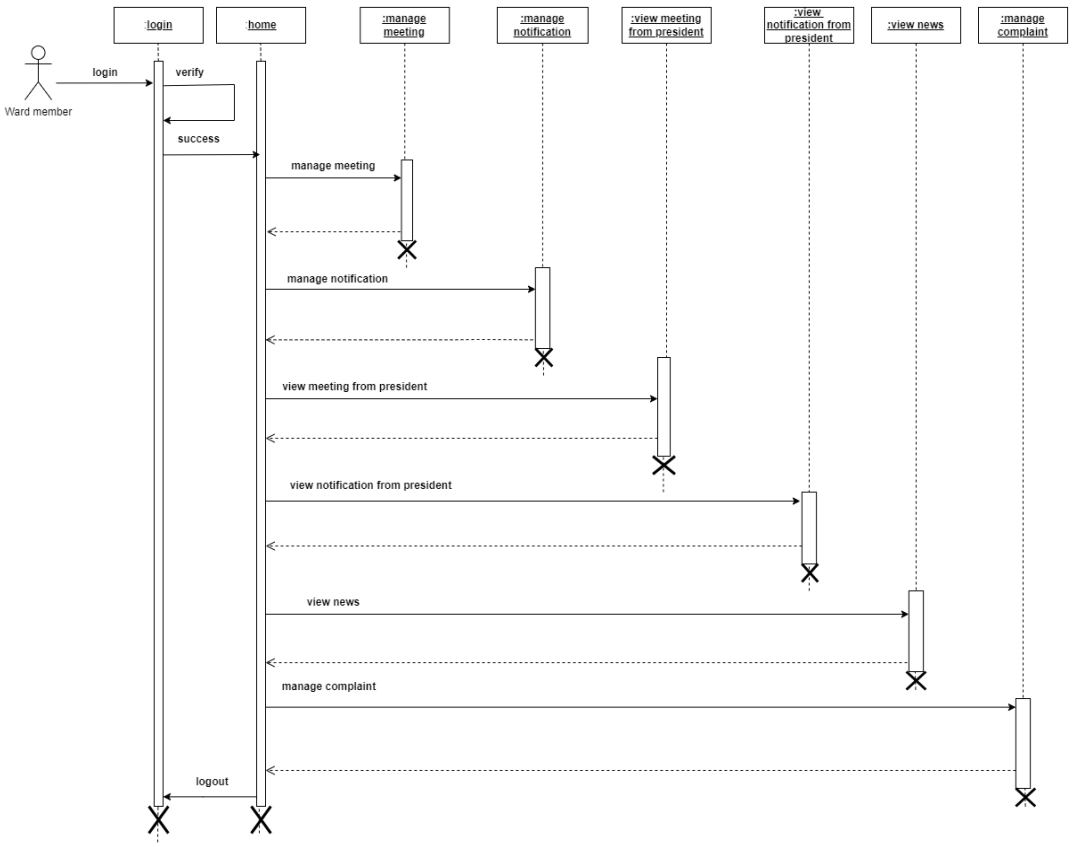
Secretary


:login
:home
:view services
:manage application
:view meeting
:view notification
:view news
Secretary
login
verify
success
view services
manage application form accepted by staff
view meeting
view notification
view news
logout

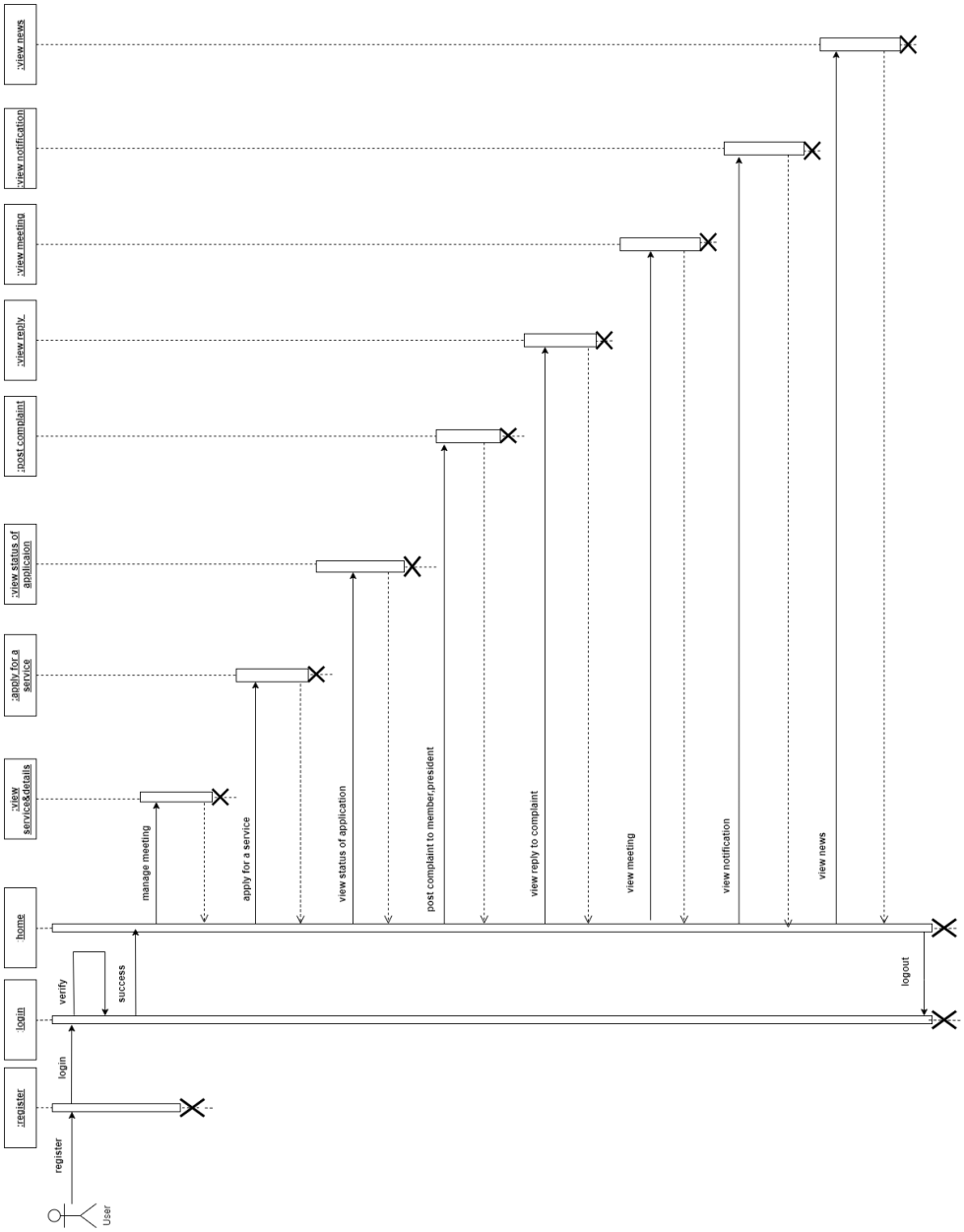
Staff


:login
:home
:add services
:add application
:manage application form from user
:view notification
:view news
:view meeting
Staff
login
verify
success
add services
add application for services
manage application form from users
view notification
view news
view meeting
logout

Ward member


:login
:home
:manage meeting
:manage notification
:view meeting from president
:view notification from president
:view news
:manage complaint
Ward member
login
verify
success
manage meeting
manage notification
view meeting from president
view notification from president
view news
manage complaint
logout

User


:register
:login
:home
:view service&details
:apply for a service
:view status of application
:post complaint
:view reply
:view meeting
:view notification
:view news
User
register
login
verify
success
manage meeting
apply for a service
view status of application
post complaint to member,president
view reply to complaint
view meeting
view notification
view news
logout

SYSTEM DEVELOPMENT

7. SYSTEM DEVELOPMENT

System development is a series of operations to manipulate data to produce output from a computer system. The principal activities performed during the development phase can be divided into two major related sequences.

- External system development
- Internal system development

The major external system activities are:

- Implementation
- Planning
- Equipment acquisition
- Installation

7.1 Coding

The purpose of code is to facilitate the identification and retrieval of items of information. A code is an ordered collection of symbols designed to provide unique identification of an entity or an attribute. Code also shows interrelationship among different items. Codes are used to identify, access, sort, matching records. The code ensures that only one value of code with a single meaning is applied to give an entity or attribute as described in various ways.

Key Features of MongoDB:

Document Model: MongoDB stores data in flexible, schema-less documents called BSON (Binary JSON) documents. BSON documents are hierarchical, allowing nested structures and arrays. This flexibility allows developers to easily evolve the data model over time.

Scalability and Performance: MongoDB is designed to scale horizontally across multiple servers and handle large amounts of data. It supports automatic sharding, which distributes data across multiple servers, enabling horizontal scalability and improved performance.

Replication and High Availability: MongoDB provides built-in replication capabilities, allowing data to be replicated across multiple nodes. This ensures high availability and data durability by maintaining multiple copies of data in case of hardware failures or network issues.

Querying and Indexing: MongoDB supports rich and expressive query capabilities, including queries on embedded documents and arrays. It offers various query operators and aggregation framework for performing advanced data analysis and manipulation. Indexing can be applied to improve query performance.

Flexibility: MongoDB's flexible data model allows developers to work with evolving requirements and accommodate changes to the data structure easily. It does not enforce rigid schemas, allowing fields to be added or modified without affecting existing data.

Ad hoc Queries: MongoDB supports ad hoc queries, allowing developers to query the database without the need to pre-define specific database views or schemas. Geospatial and Full-Text Search: MongoDB provides geospatial indexing and querying capabilities, enabling efficient storage and querying of geospatial data. It also offers full-text search functionality to perform text-based searches on data stored in the database.

Rich Ecosystem: MongoDB has a vibrant and extensive ecosystem with a wide range of libraries, drivers, and tools available for different programming languages and platforms. This makes it easier to integrate MongoDB into existing software projects and frameworks.

Node JS

Node js is an open-source, cross-platform JavaScript runtime environment that enables developers to build scalable and high-performance applications. It is built on top of the V8 JavaScript engine used by Google Chrome and provides an event- driven, non-blocking I/O model that makes it well-suited for real-time web applications. Node.js enables developers to write server-side applications using JavaScript, which is a popular and widely-used programming language on the web. It has a vast ecosystem of third-party packages and libraries that can be easily installed using the Node Package Manager (NPM). Node js applications can be run on various platforms such as Windows, Mac, and Linux.

Express JS

Express.js is a minimal and flexible Node.js web application framework that provides a set of robust features for building web and mobile applications. It is one of the most popular and widely-used frameworks for Node.js, and is known for its simplicity and ease of use. Express.js provides a set of features for developing server side web applications, including routing, middleware support, template engines,and much more. It also provides an easy-to-use API for interacting with databases such as MongoDB and MySQL, and supports a variety of templating engines, such as Pug, Handlebars, and EJS.

Mongo DB

MongoDB is a popular document-oriented NoSQL database system that allows developers to store and manage large amounts of data in a flexible and scalableway. It is an open-source database that uses JSON-like documents with optional schemas, which makes it easy to work with and suitable for a variety of use cases. One of the key benefits of MongoDB is its ability to scale horizontally. This means that developers can add new servers to their database cluster as the amount of data or traffic increases, which allows the database to handle more requests and ensures that it can continue to perform well even as the application grows.

Session

Session refers to an authenticated state or login session with the Node Package Manager (npm). It represents a user's authorization and access privileges to perform various actions, such as installing packages, publishing packages, or managing user account settings, using the npm command line interface (CLI). The session typically persists until the user explicitly logs out or it expires due to inactivity. During an npm session, the user can leverage their credentials to interact with the npm registry and carry out package management tasks within their development environment.

Express-fileupload

Express-fileupload is an npm package that provides middleware for handling file uploads in Express.js applications. It simplifies the process of receiving and processing files submitted as part of form data or through API endpoints. The middleware automatically parses and extracts uploaded files from the request, making them easily accessible for further processing. It allows you to handle file uploads seamlessly within your Express.js routes and perform operations such as saving files to disk, validating file types, resizing images, and more. By using express-fileupload, you can efficiently handle file upload functionality in your Express.js application without having to implement the parsing and handling logic from scratch.

Express Handlebars Express Handlebars, also known as "express-handlebars," is a templating engine for the Express.js framework. It allows you to render dynamic HTML pages by combining templates with data. Handlebars provides a syntax that enables the insertion of variables, expressions, and logic into templates. It simplifies the process of generating HTML content on the server side and sending it to the client. Express Handlebars offers features like template layouts, partials, and helpers, which enhance code reusability and flexibility. It is a popular choice for developers working with Express.js to create dynamic and data-driven web applications.

SYSTEM TESTING AND IMPLEMENTATION

8. SYSTEM TESTING AND IMPLEMENTATION

Testing is vital to the success of the system. It makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved in this project. It is the stage of implementation, which ensures that the system works accurately and effectively before the live operation commences. It is a confirmation that all are correct and an opportunity to show users that the system must be tested and show that the system will operate successfully and produce expected results under expected conditions. Software testing is a crucial element of software quality assurance and represents the unlimited review of specification, design and coding. Testing represents an interesting anomaly for the software. During the earlier definition and development phase, it was attempted to build the software from an abstract concept to implement.

Testing is a set of activities that can be planned in advance and conducted. Systematically, this is aimed at ensuring that the system works accurately and efficiently before live operations commences

8.1 Types of testing

Different types of testing are

- Unit testing
- Black Box testing

Unit testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. All modules were tested individually as soon as they were completed and were checked for their correct functionality. Unit testing deals with testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. This testing is carried out during the programming stage itself. In this testing step

each Module is found to be working satisfactorily as regard to the expected output from the module.

Black Box Testing

In black-box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or specifications of the program or module, and the internals of the module or the program are not considered for selection of test cases. In black-box testing, the tester only knows the inputs that can be given to the system and what output the system should give. This form of testing is also called functional or behavioural testing. The most obvious functional testing procedure is exhaustive testing. One criterion for generating test cases is to generate them randomly. There are no formal rules for designing test cases for functional testing.

8.2 Implementation

Implementation is the stage of project, when theoretical design is turned in to a working system. The most crucial stage is achieving a successful system and confidence that the new system will be work effectively. It involves careful planning, investigation of the manual system and to new system.

There are several activities involved while implementing a project:

- Careful planning.
- Investigating the current system and its constraints on implementation.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover procedure and evaluation of change over method.

The first task in implementation was the implementation planning, that is deciding on methods to be adopted. After the system was implemented successfully, training of the user was one of the most important subtasks of the developer. For this purpose, the user or system manual were prepared and handled over to the user to operate the developed system

SYSTEM MAINTENANCE

9. SYSTEM MAINTENANCE

Maintenance is making adaptation of the software for external changes (requirements changes or enhancements) and internal changes (fixing bugs). When changes are made during the maintenance phase all preceding steps of the model mustbe revisited.

There are three types of maintenance:

- Corrective (fixing bugs/error)
- Adaptive (Updates due to environment changes)
- Perfective (Enhancements, requirements changes)

Maintenance is an enigma of the system development. The definition of the software maintenance can be given describing four activities that are undertaken after the program is released for use.

The maintenance activity occurs since it is unreasonable to assume that software testing will uncover all in a large system. The second activity that contributes to the definition of maintenance occurs since rapid changes are encountered in every aspect of computing. The third activity involves recommendation for new capabilities, modification to the existing functions and general enhancements when the software is used. The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability.

FUTURE ENHANCEMENT

10. FUTURE ENHANCEMENT

A future enhancement for the Ulliyeri-e-Panchayath portal could involve integrating a payment gateway to allow residents to pay for the services they access through the platform conveniently. This would streamline the process for both users and service providers, reducing the need for cash transactions and ensuring secure and transparent transactions. Additionally, implementing features such as transparent pricing, invoicing, and digital receipts could further enhance the financial transparency of the platform and build trust among users. Furthermore, exploring partnerships with local financial institutions or mobile payment providers could expand the range of payment options available to residents, catering to varying preferences and levels of access to banking services in rural areas. Overall, integrating payment functionalities into the platform would not only enhance user experience but also contribute to the sustainability and scalability of the project by potentially generating revenue to support its continued development and operation.

CONCLUSION

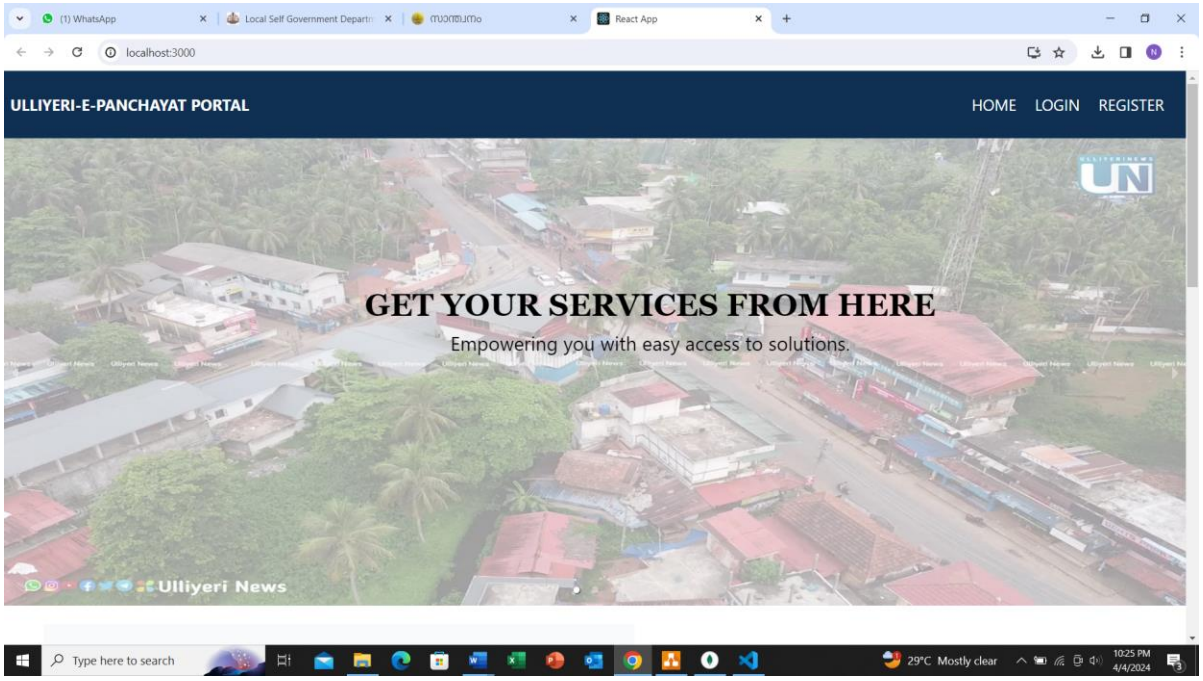
11. CONCLUSION

In conclusion, the Ulliyeri-e-Panchayath portal represents a significant step towards modernizing and digitizing Panchayat services, with a focus on enhancing accessibility, efficiency, and transparency in rural governance. By leveraging technology, the project aims to empower local communities, improve communication channels, and facilitate transparent decision-making processes. Through the development of a user-friendly platform, residents will be able to access essential services, participate in local governance, and bridge the digital divide in rural areas. Additionally, the project's goal to create a platform for locating local home service workers further emphasizes its commitment to meeting the diverse needs of rural communities.

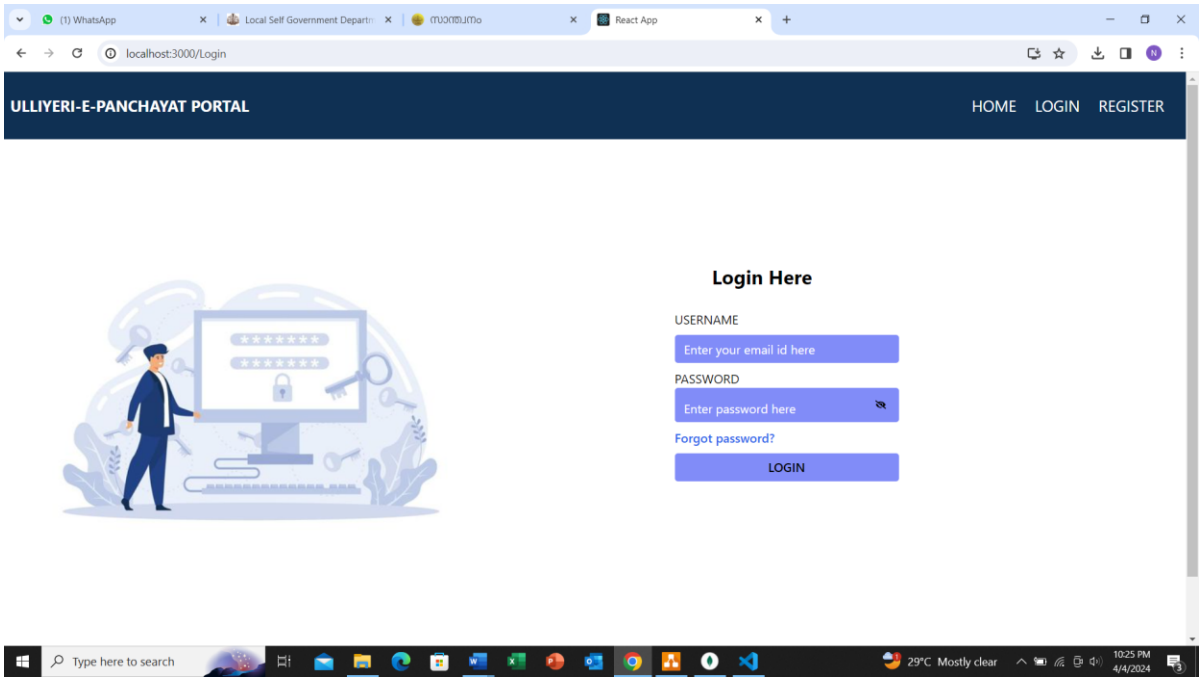
APPENDIX

12. APPENDIX

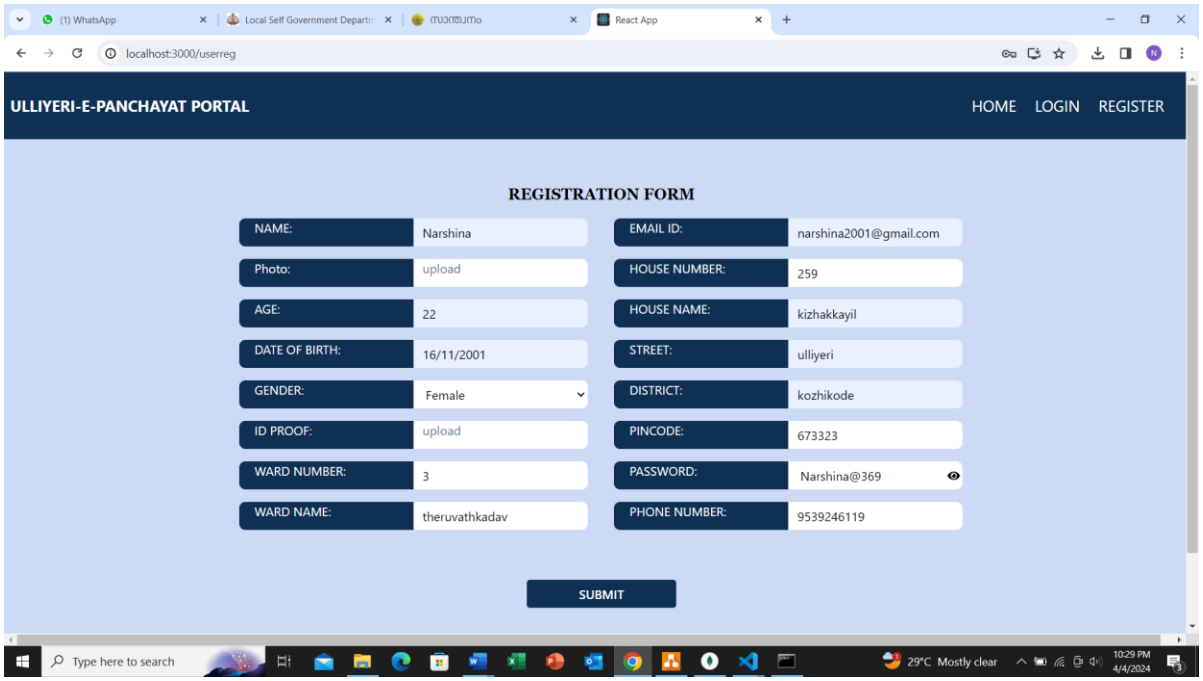
Home page



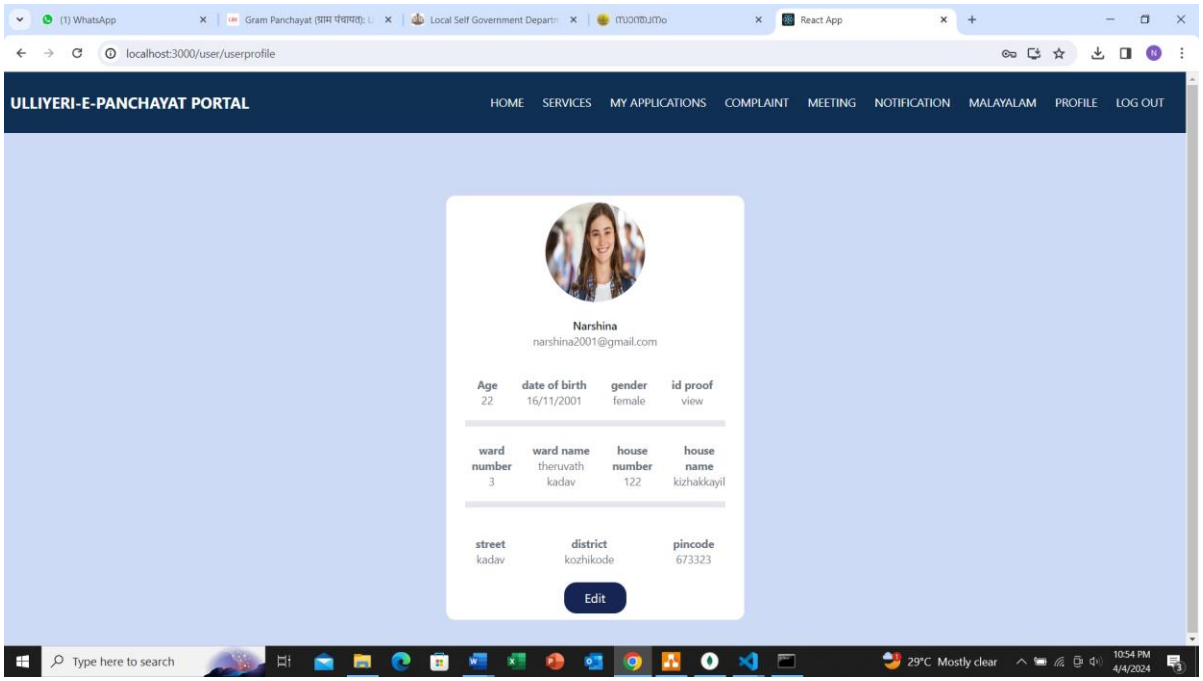
Login



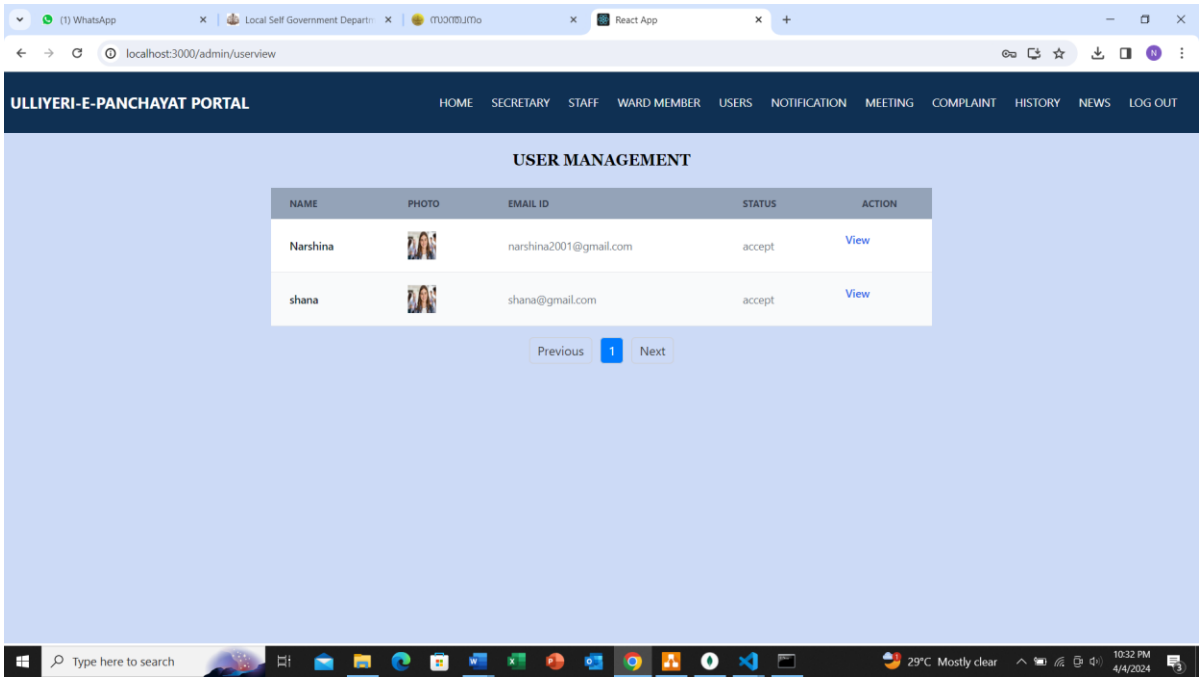
User registration



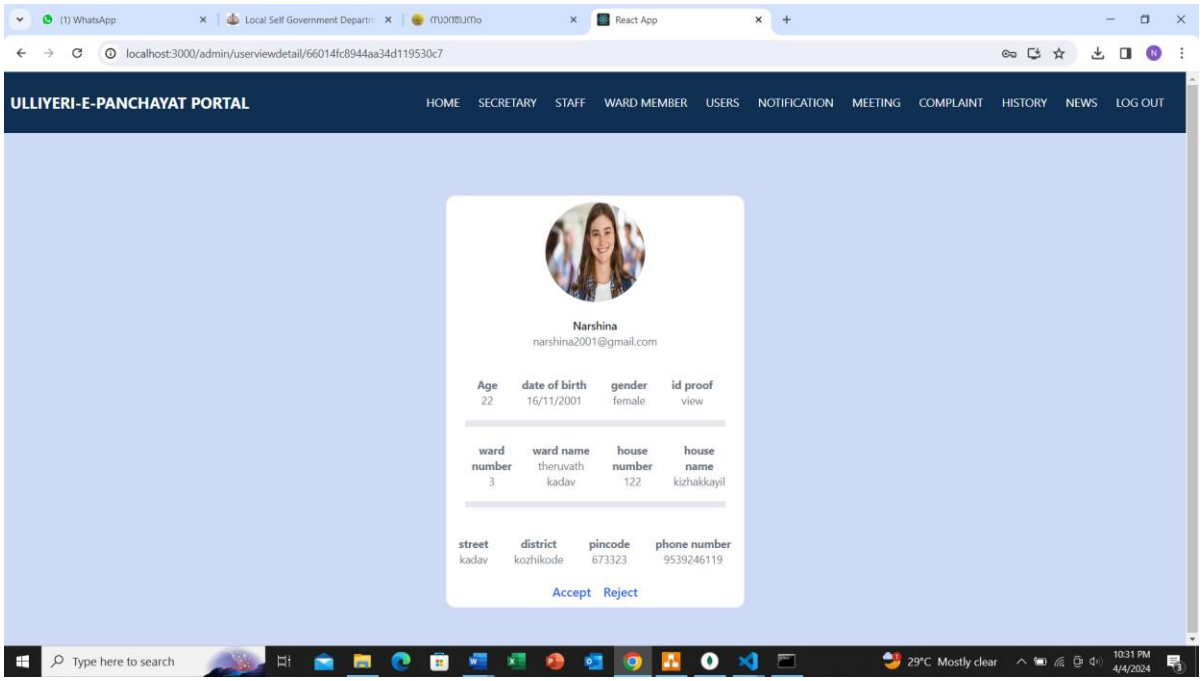
User profile



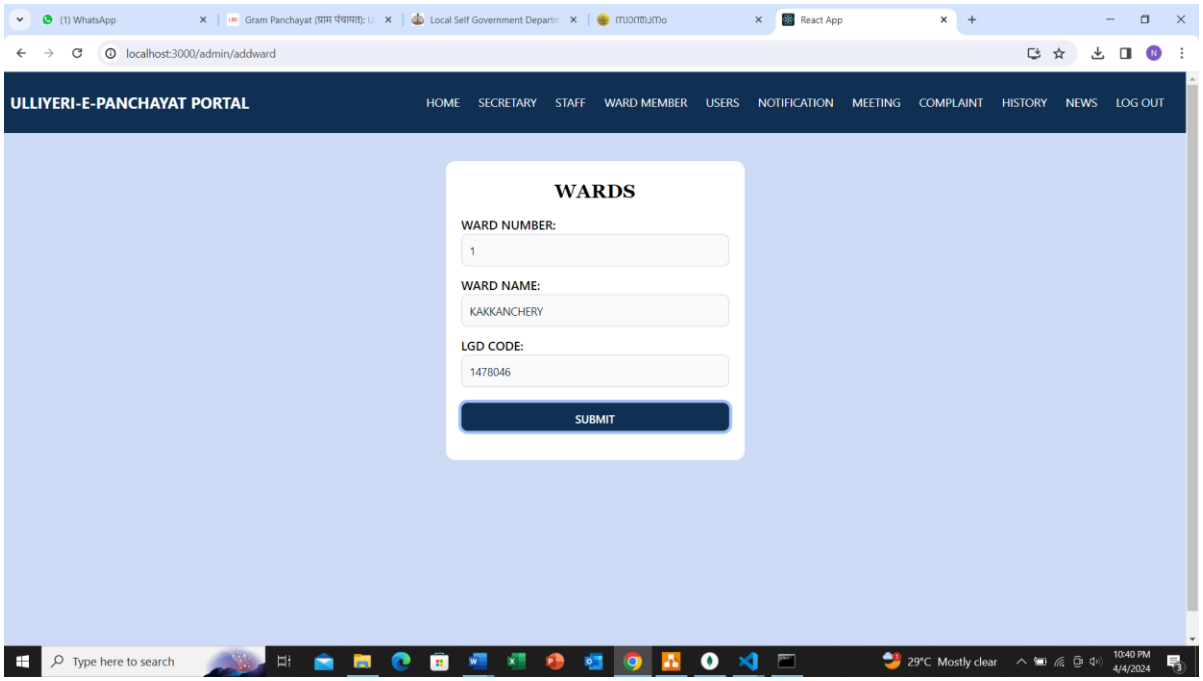
View users



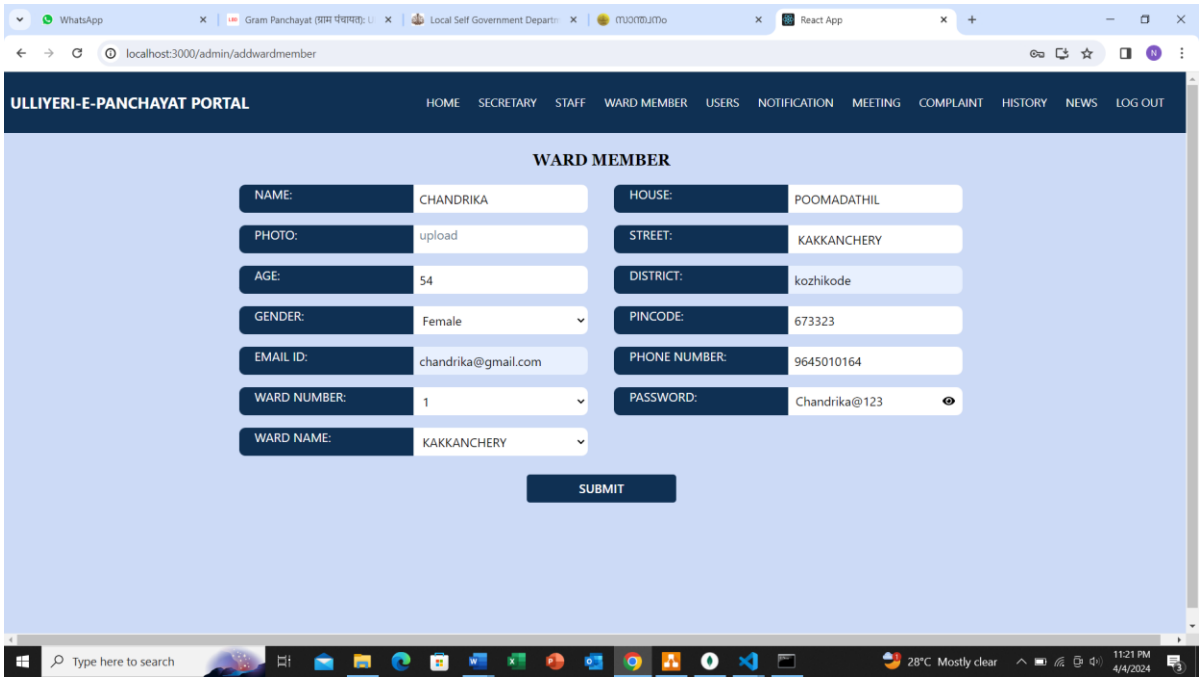
User management



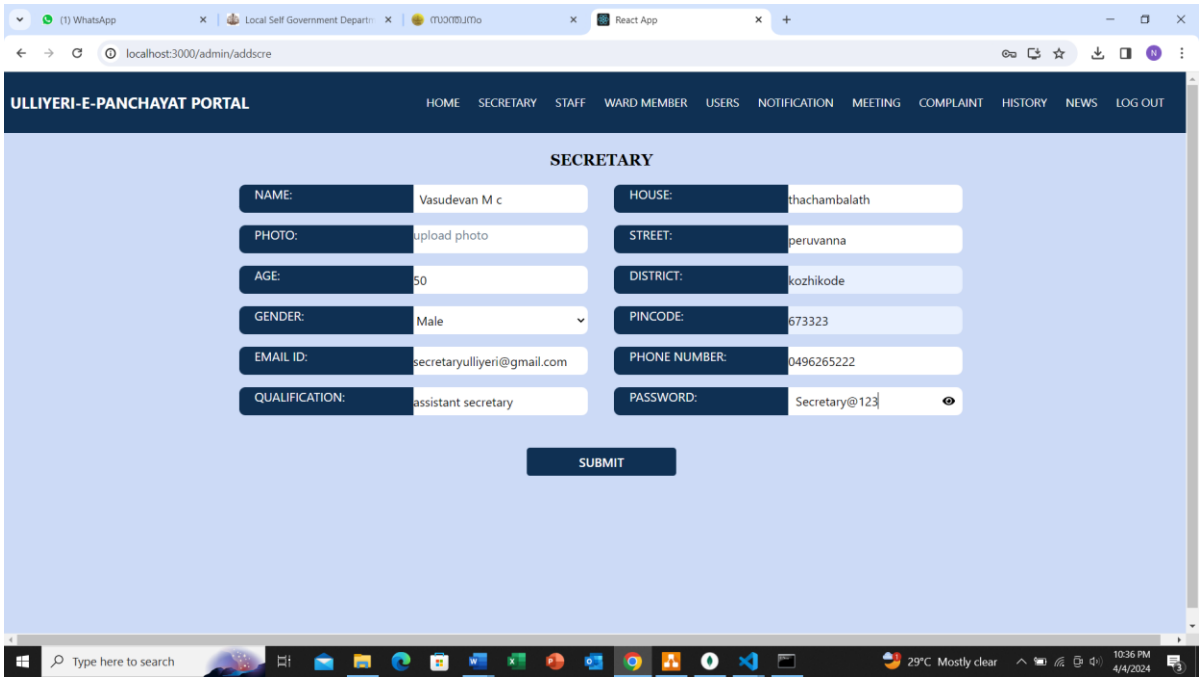
Add ward



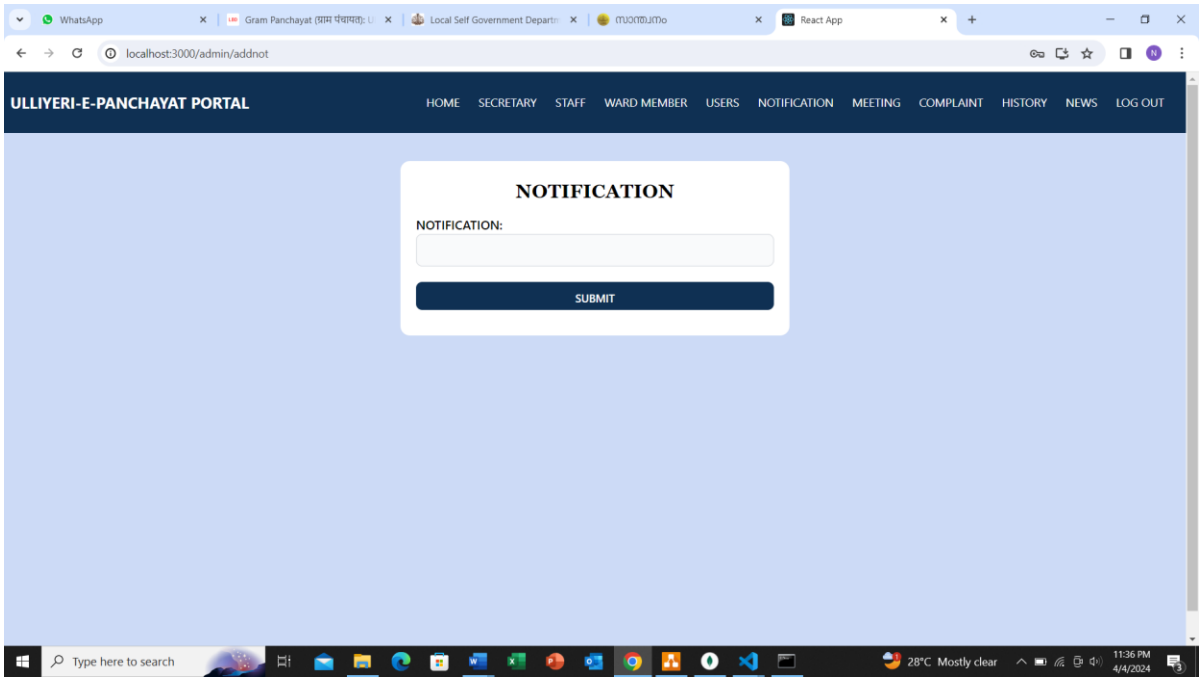
Add ward member



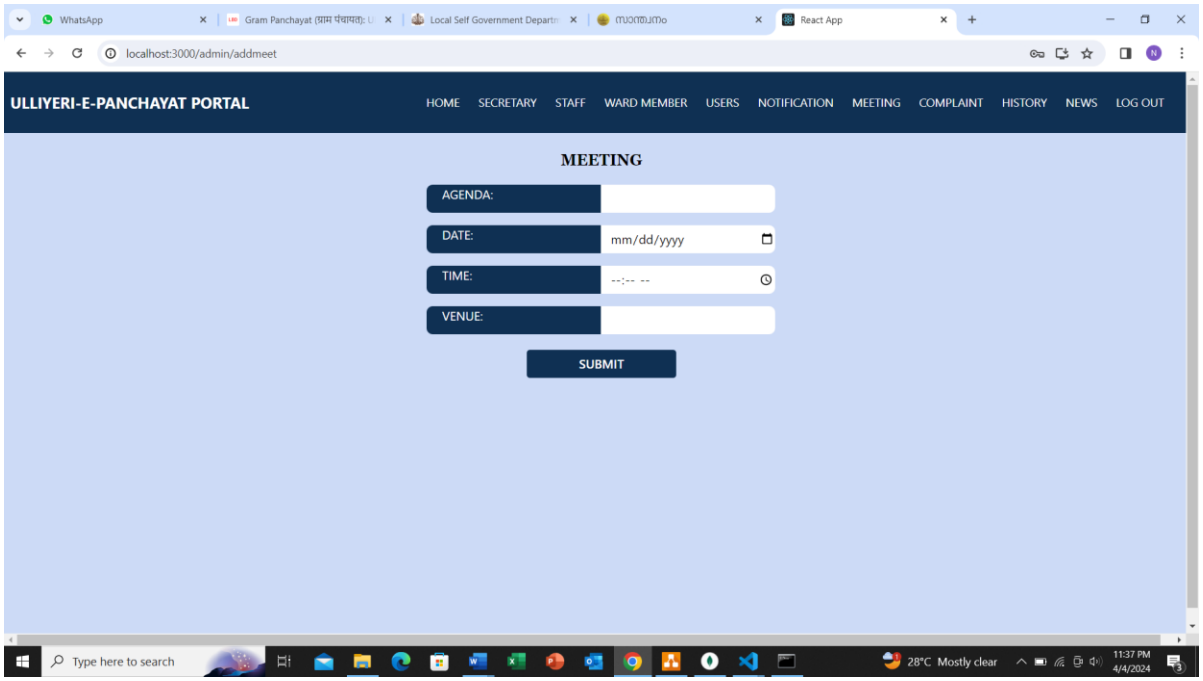
Add secretary



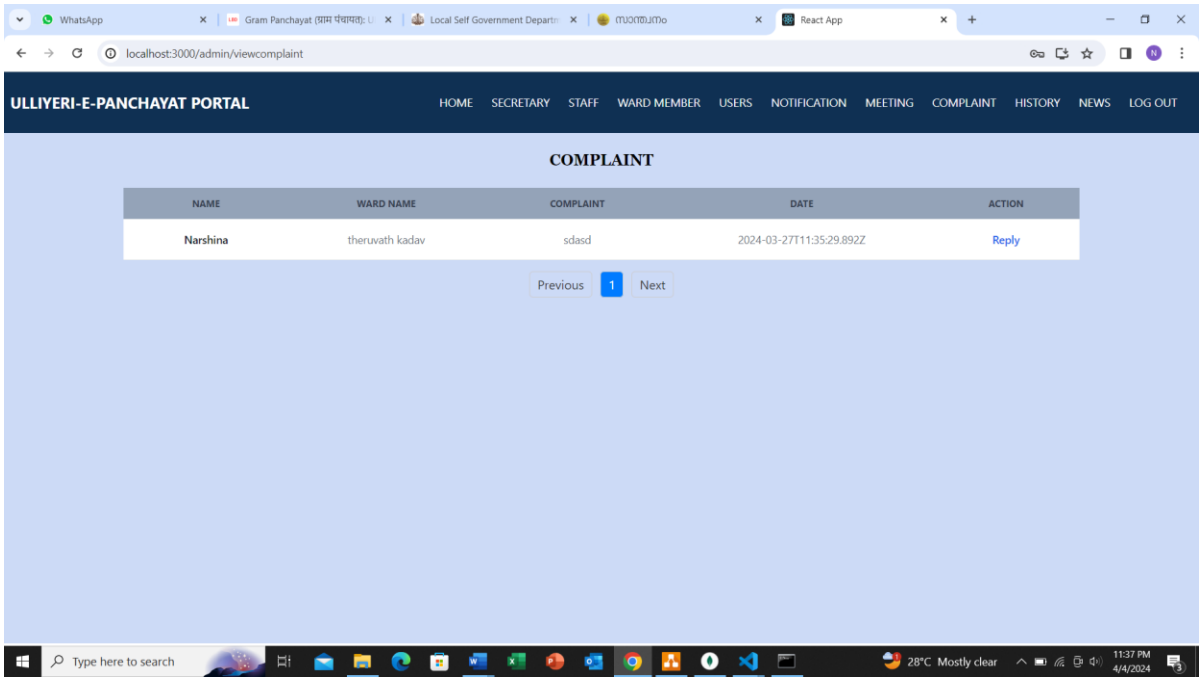
Add notification



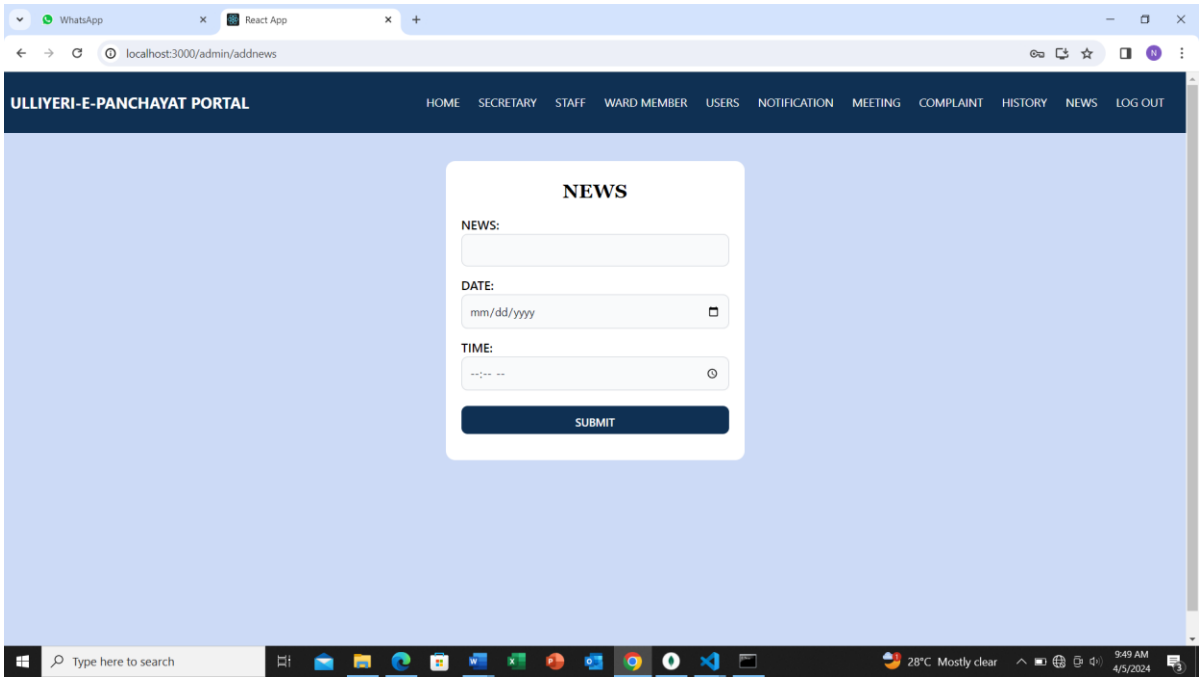
Add meeting



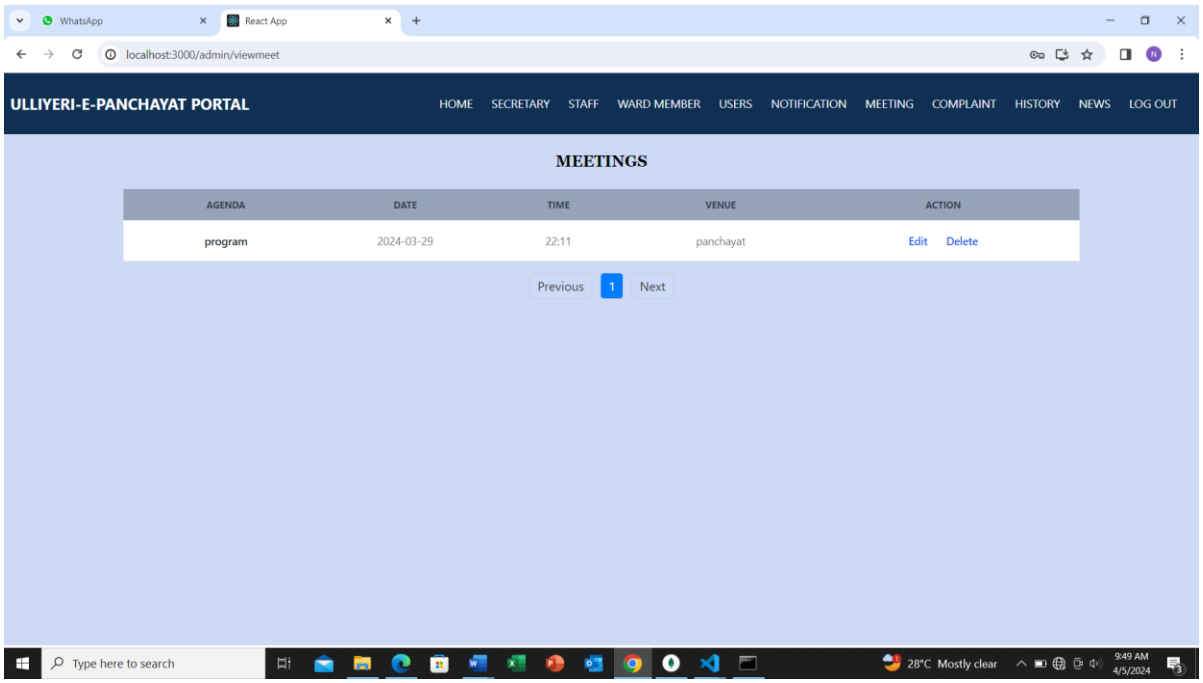
View meeting



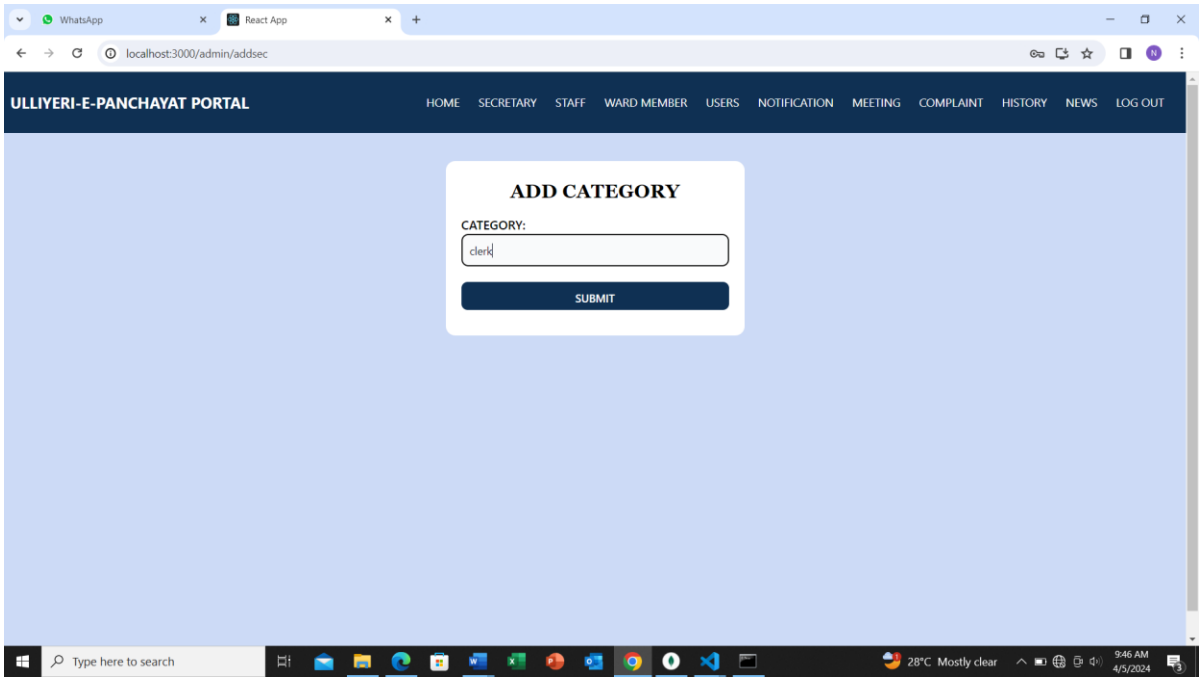
Add news



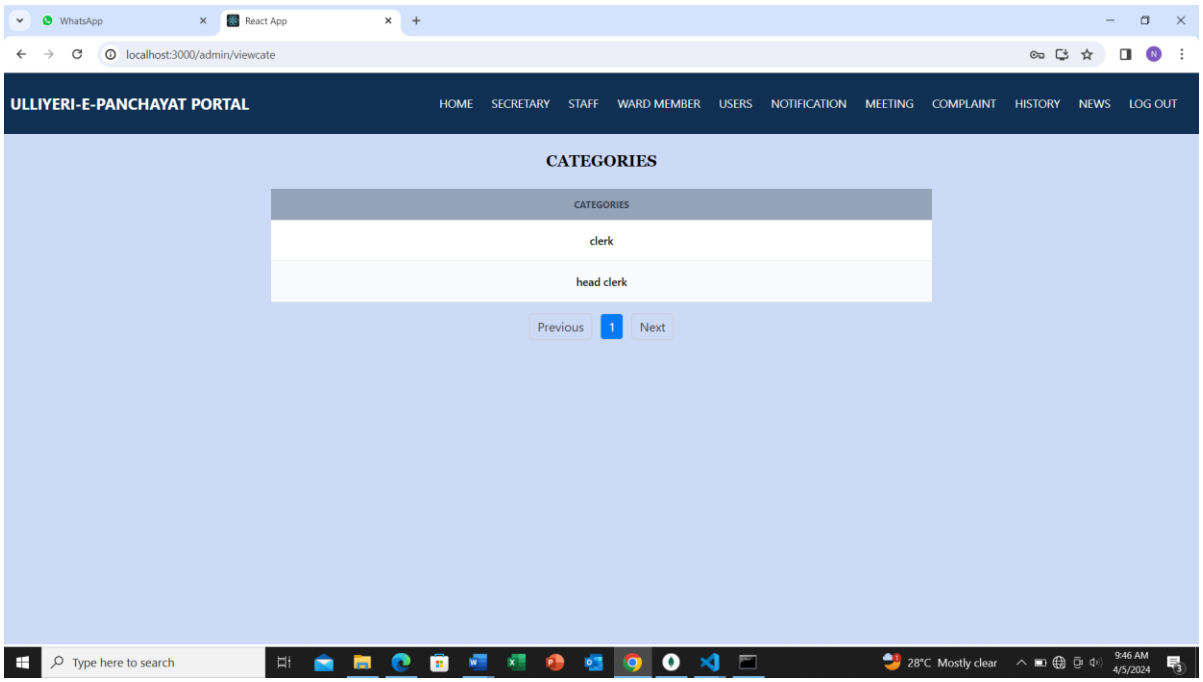
View meeting



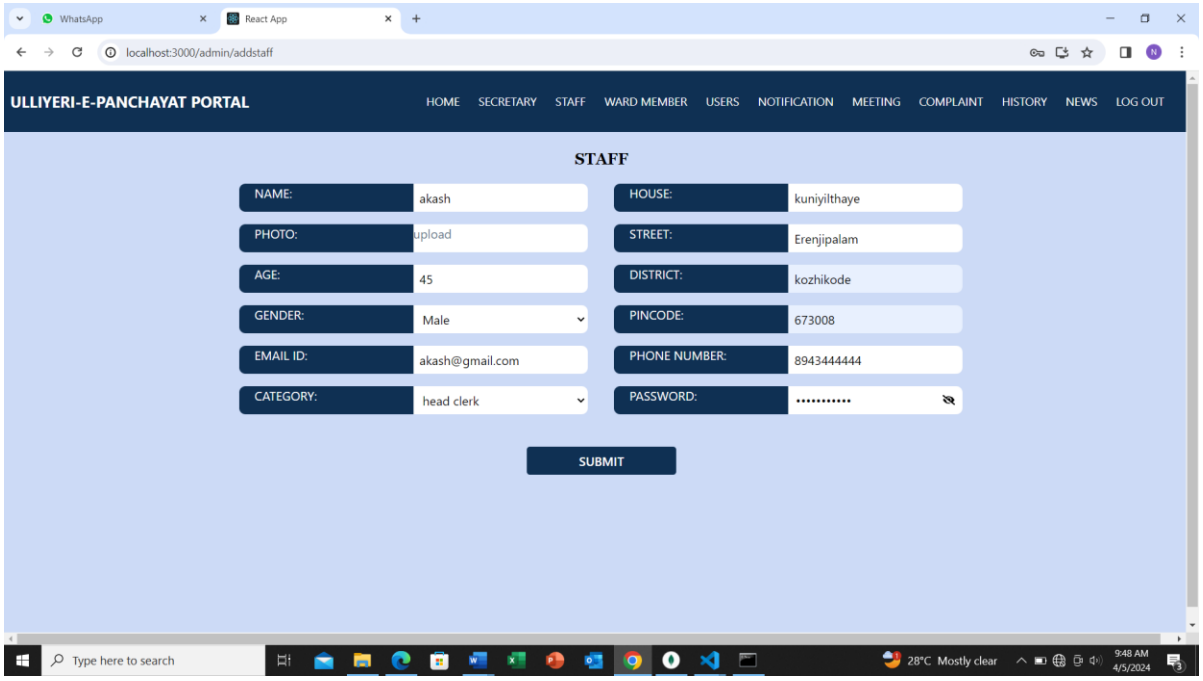
Add category



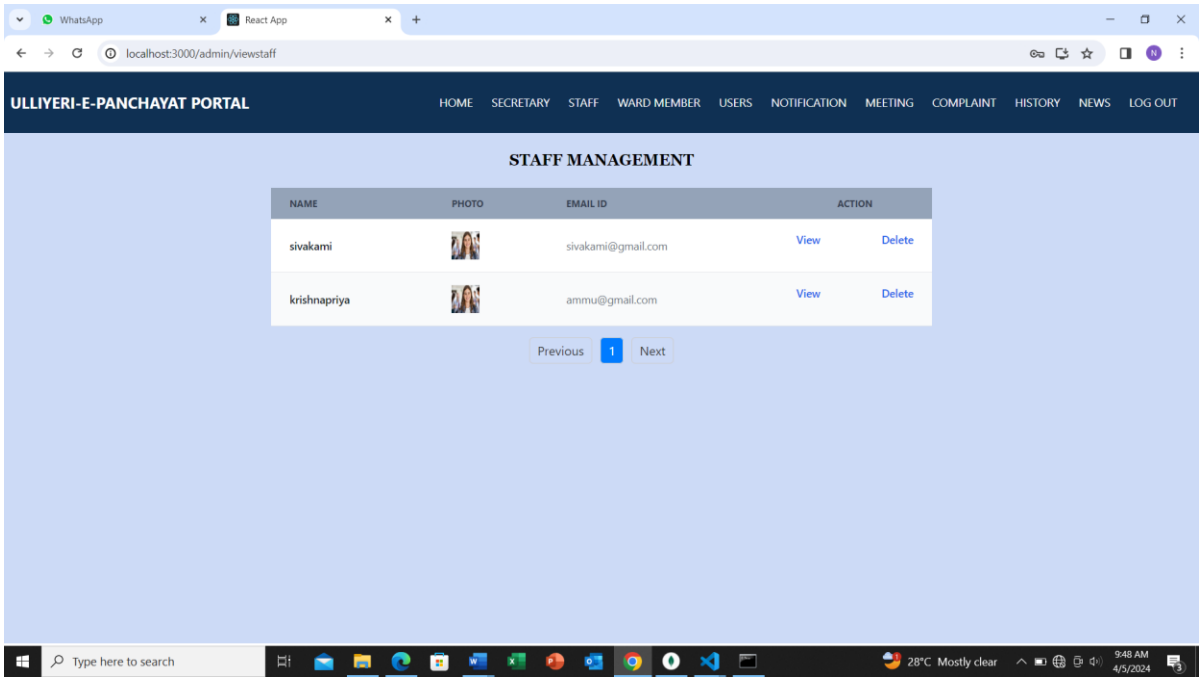
View category



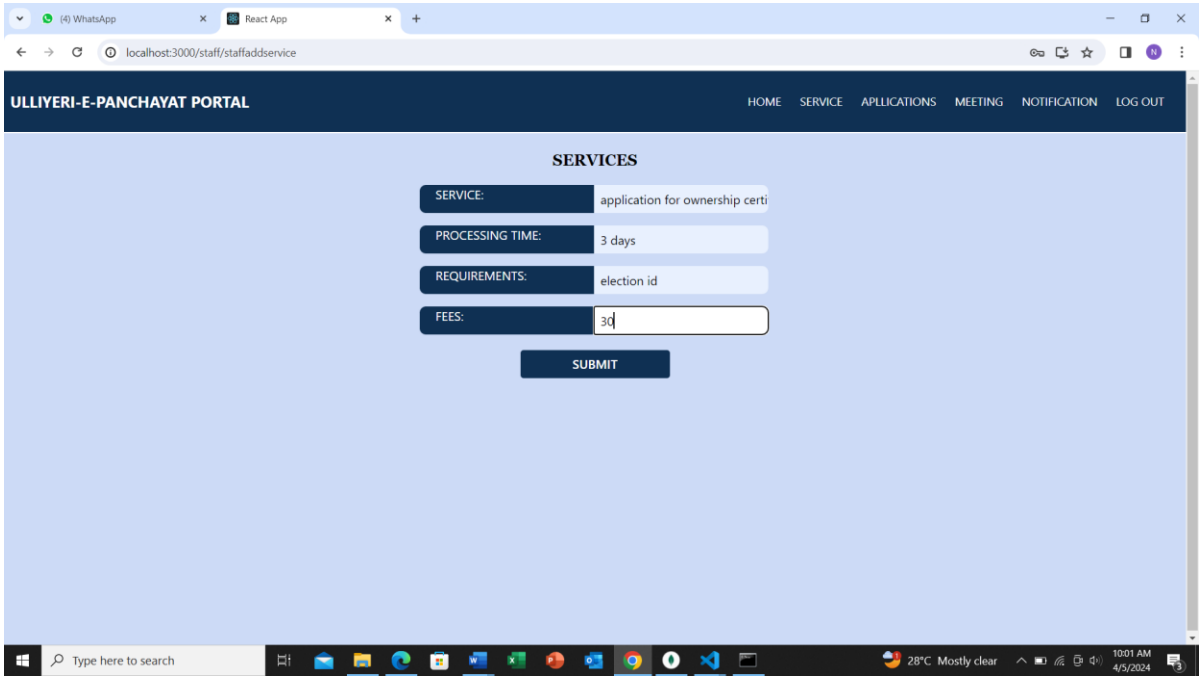
Add staff



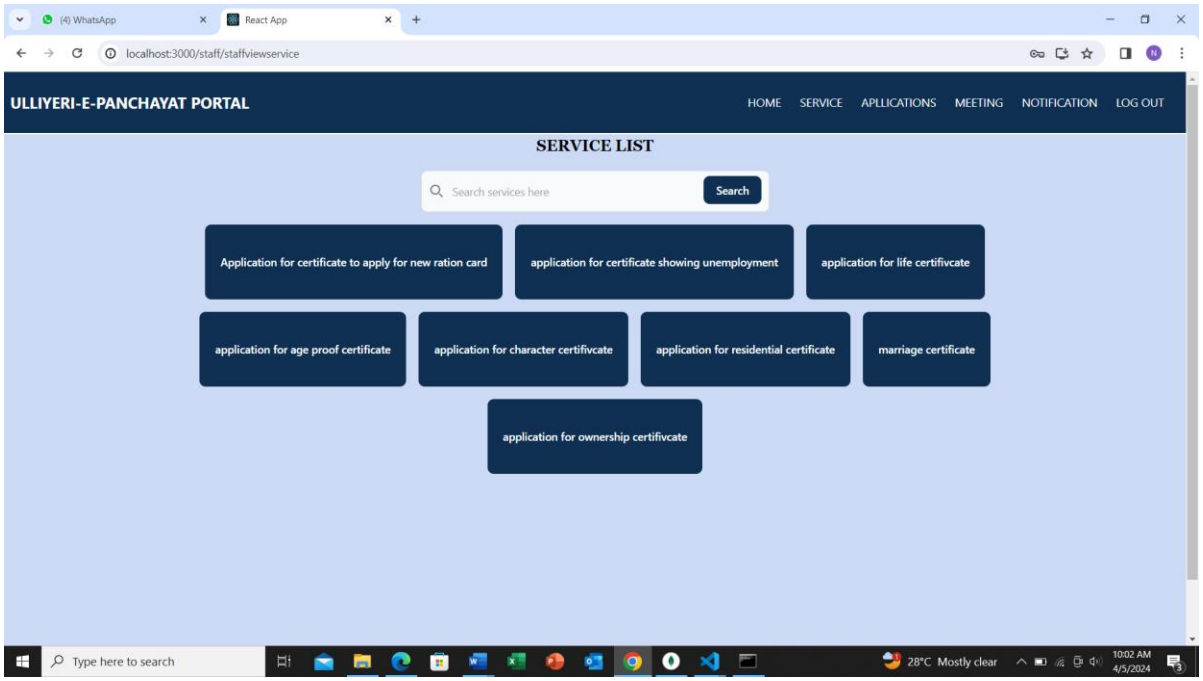
View staff



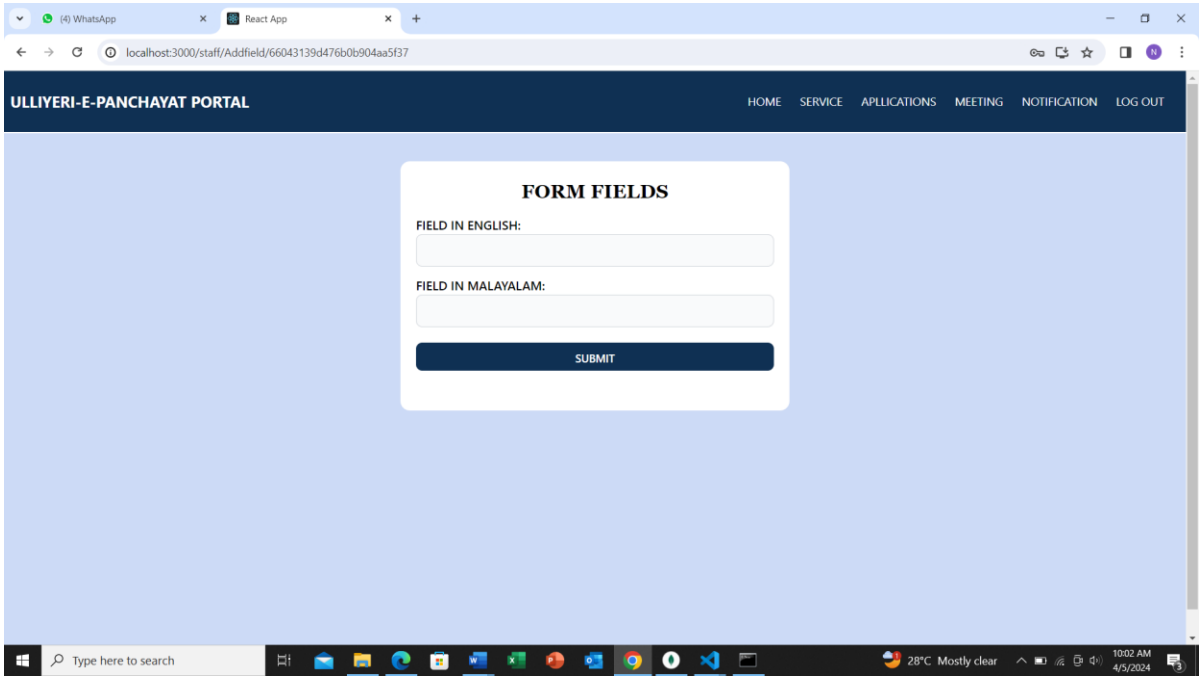
Add services



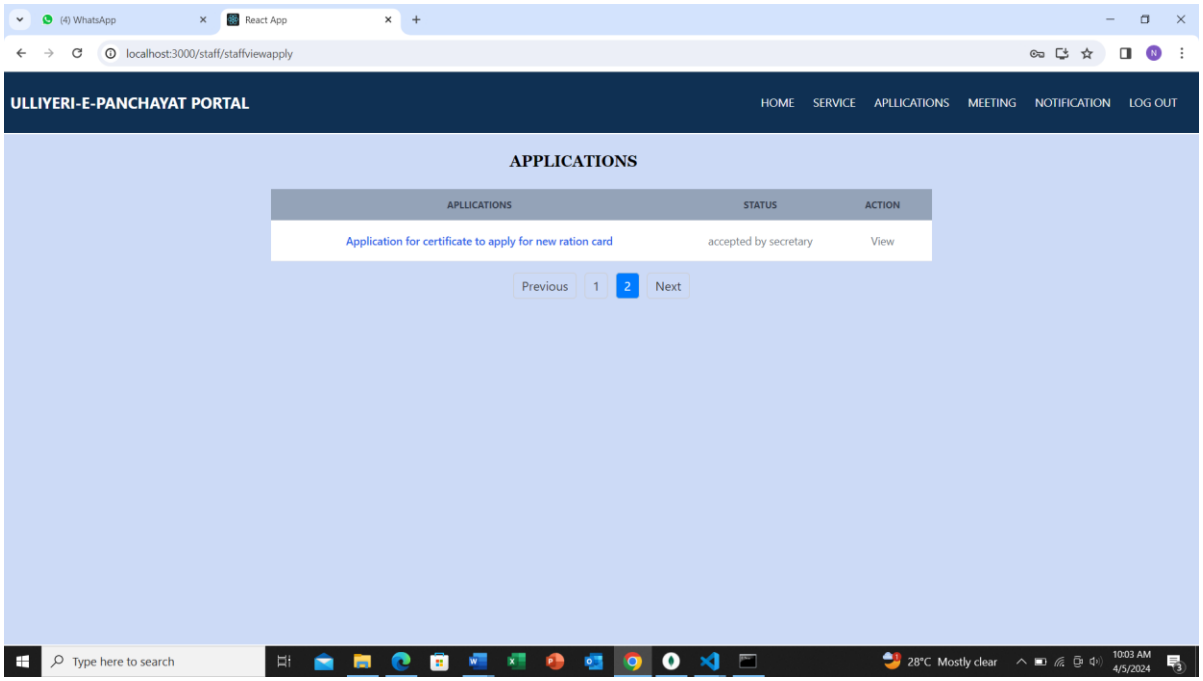
View services



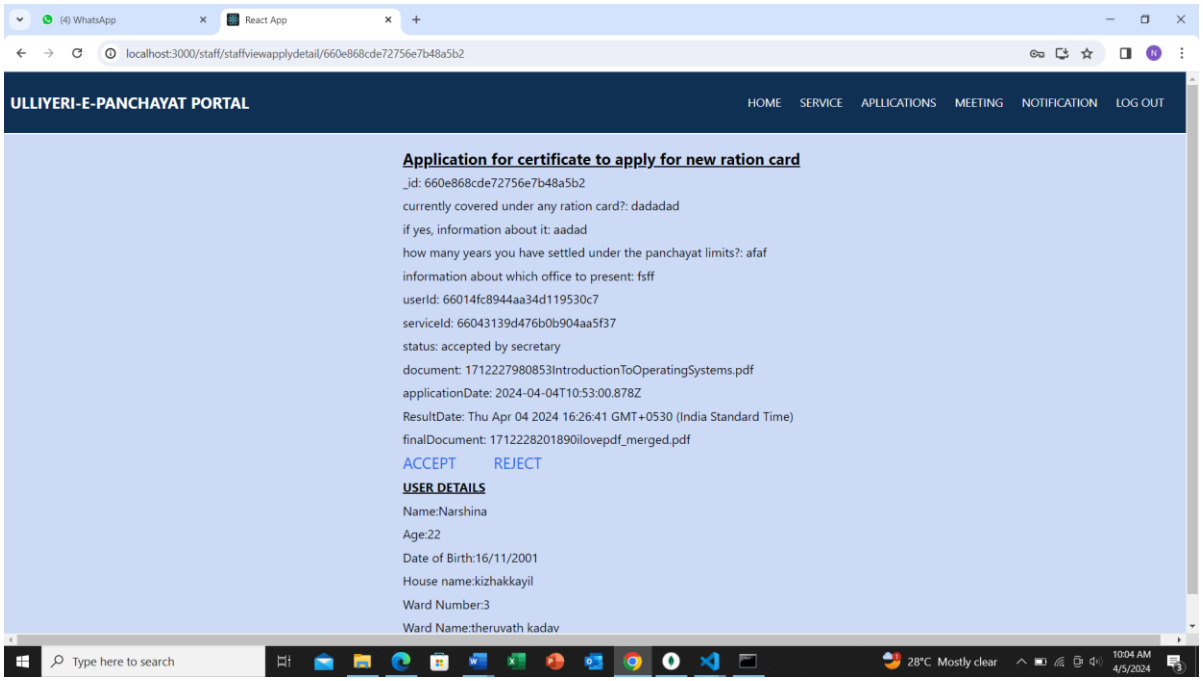
Form fields



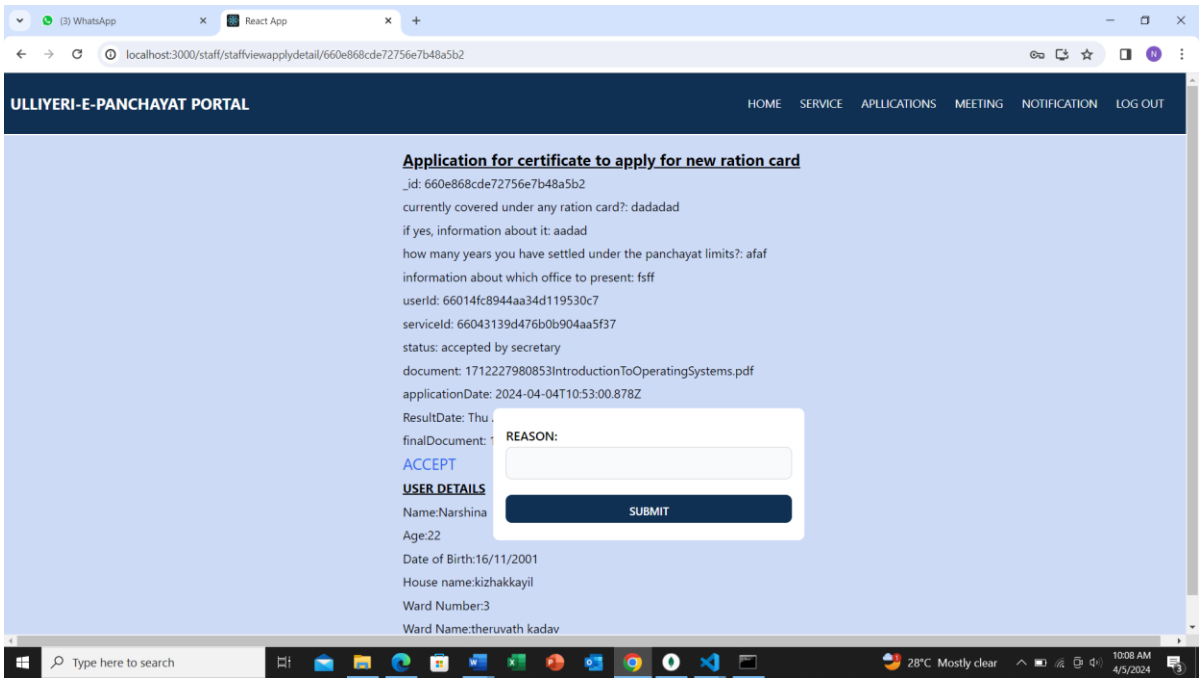
View applications



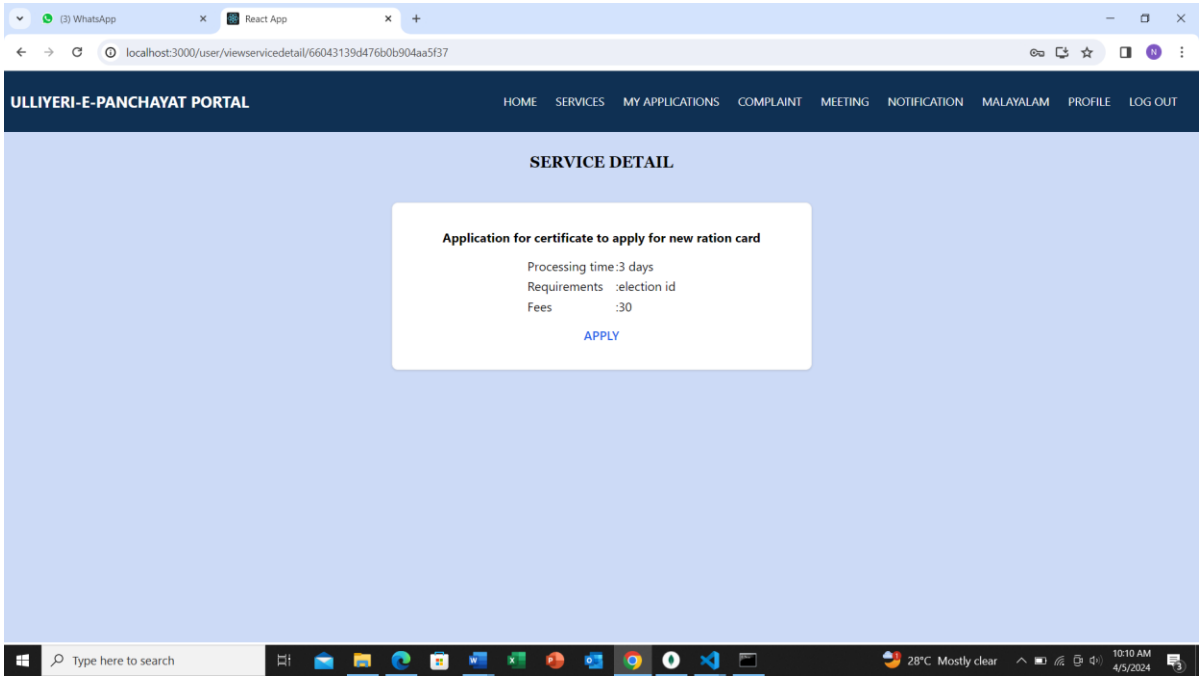
View application in detail



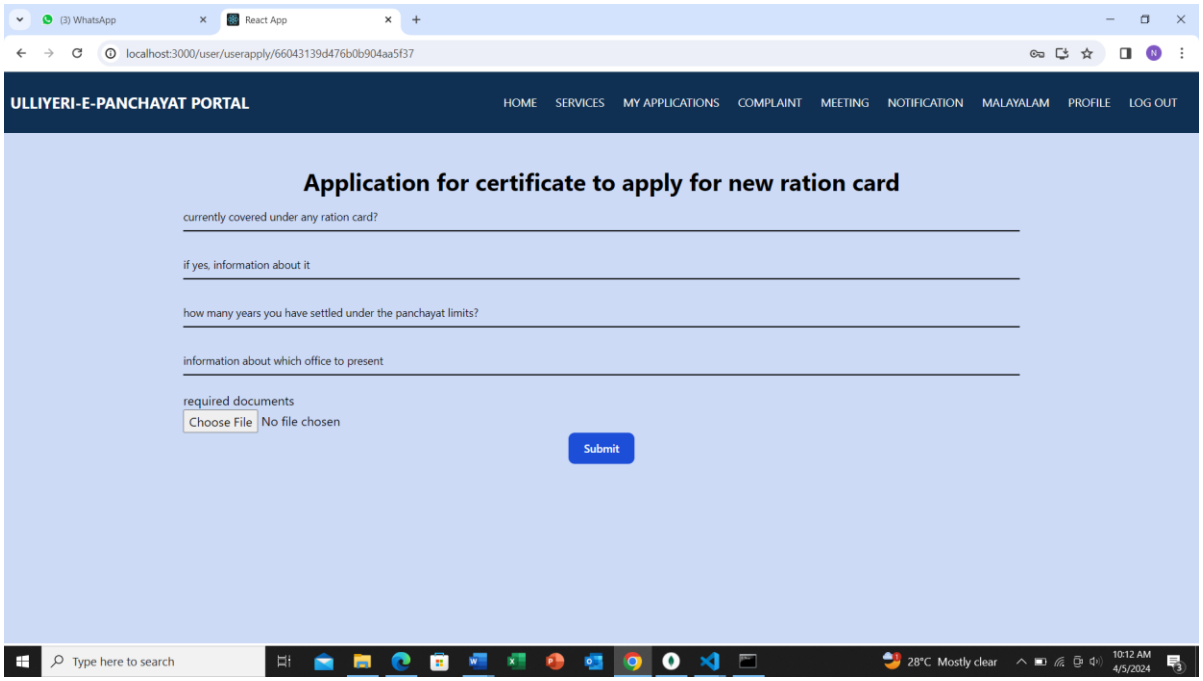
Add rejected reason



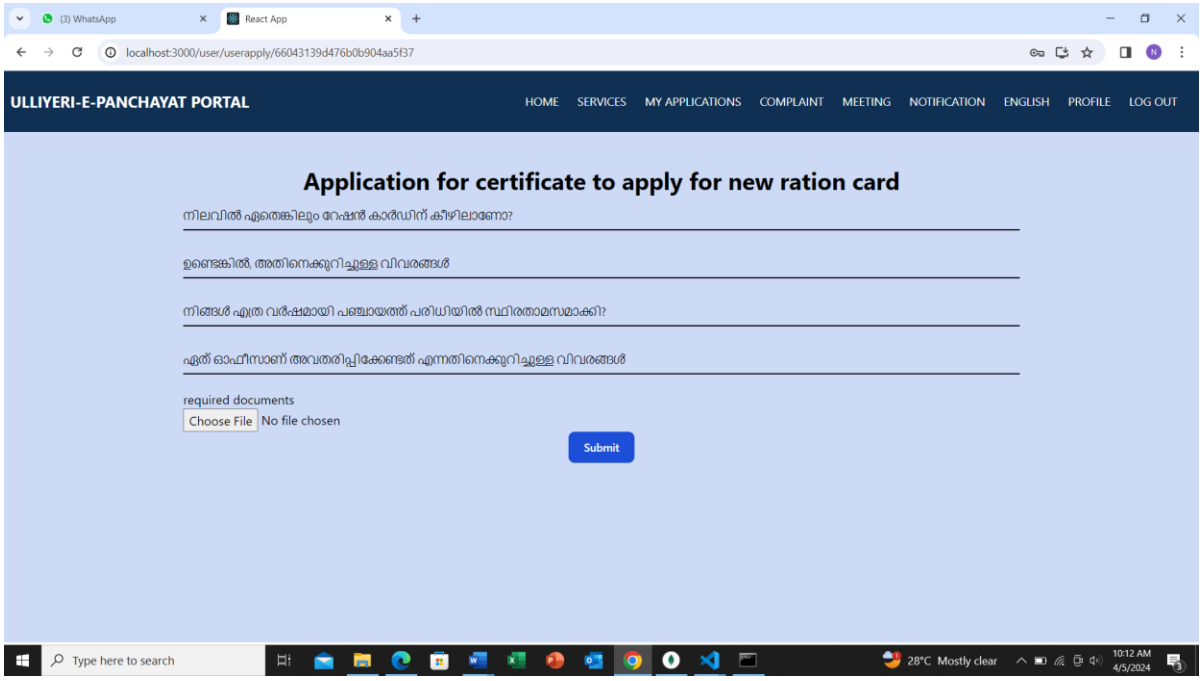
View service detail



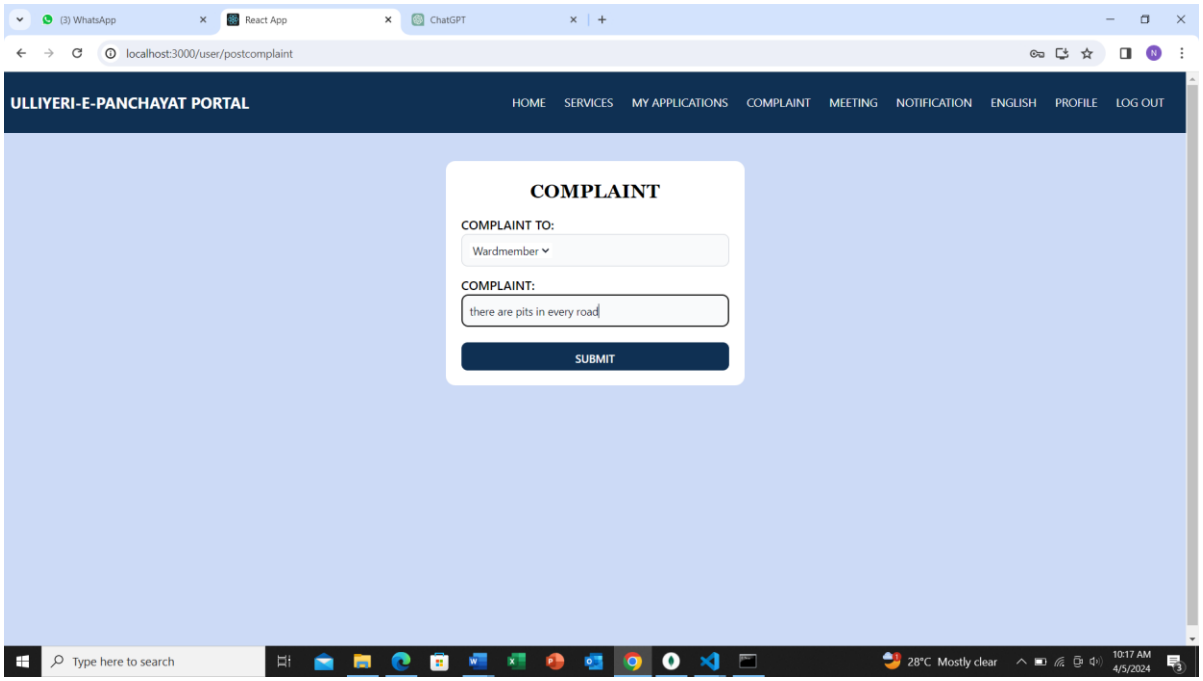
View application form in English



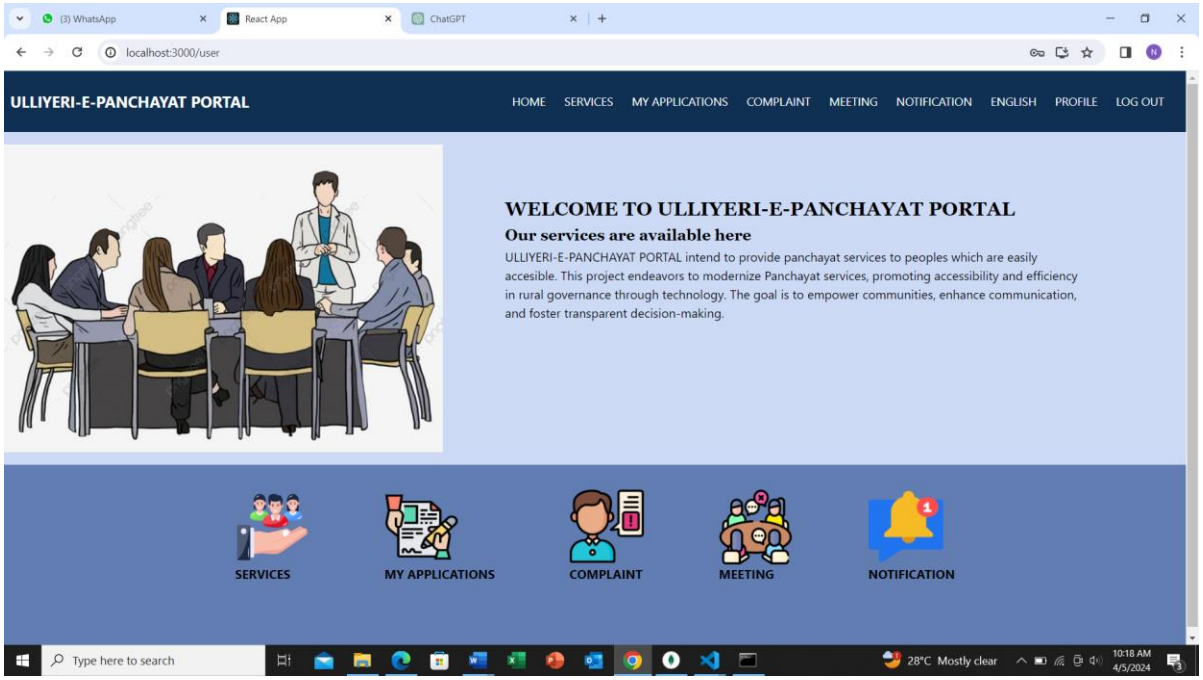
View application in Malayalam



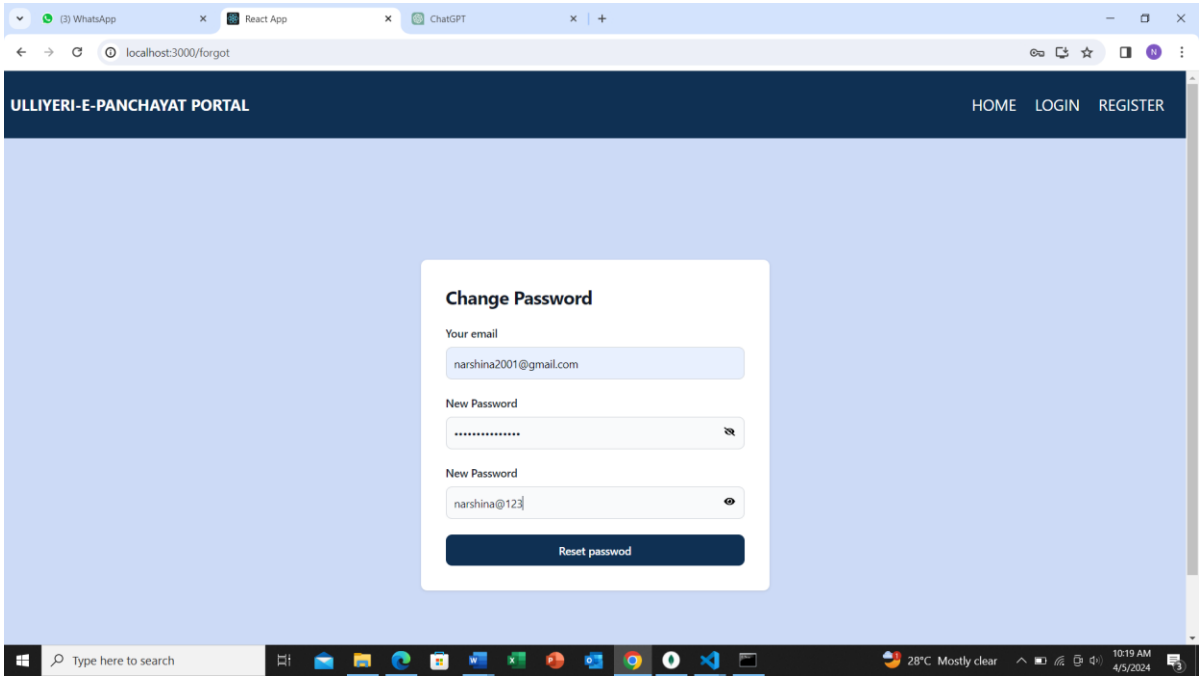
complaint



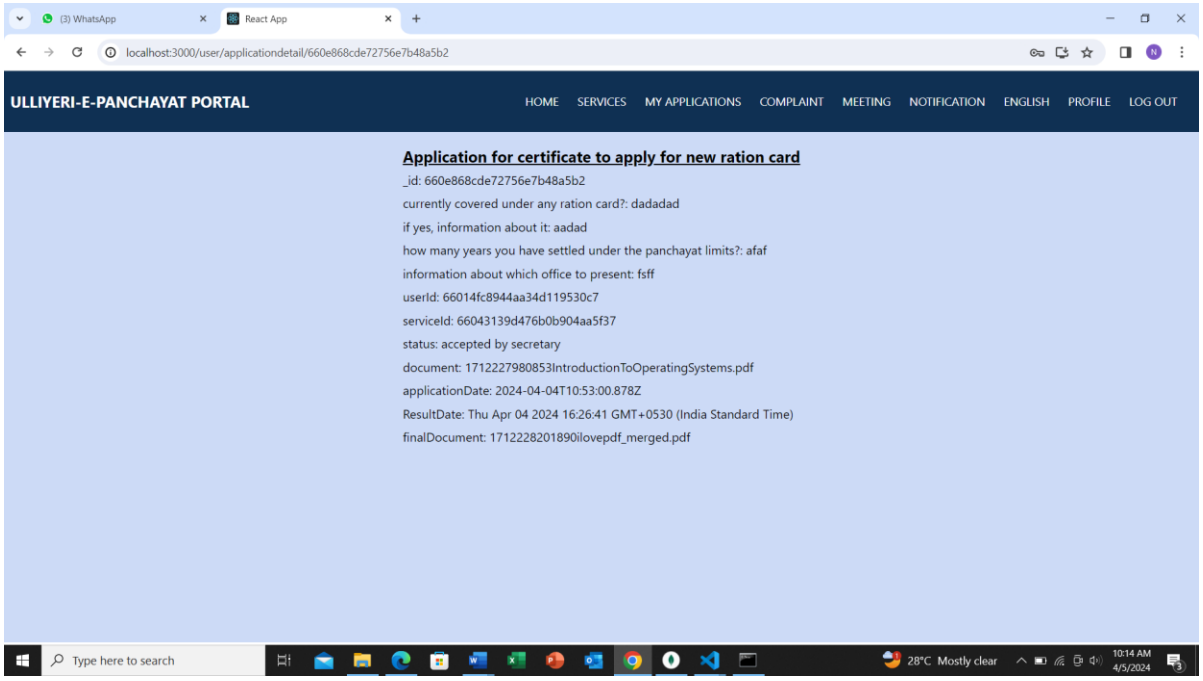
User home page



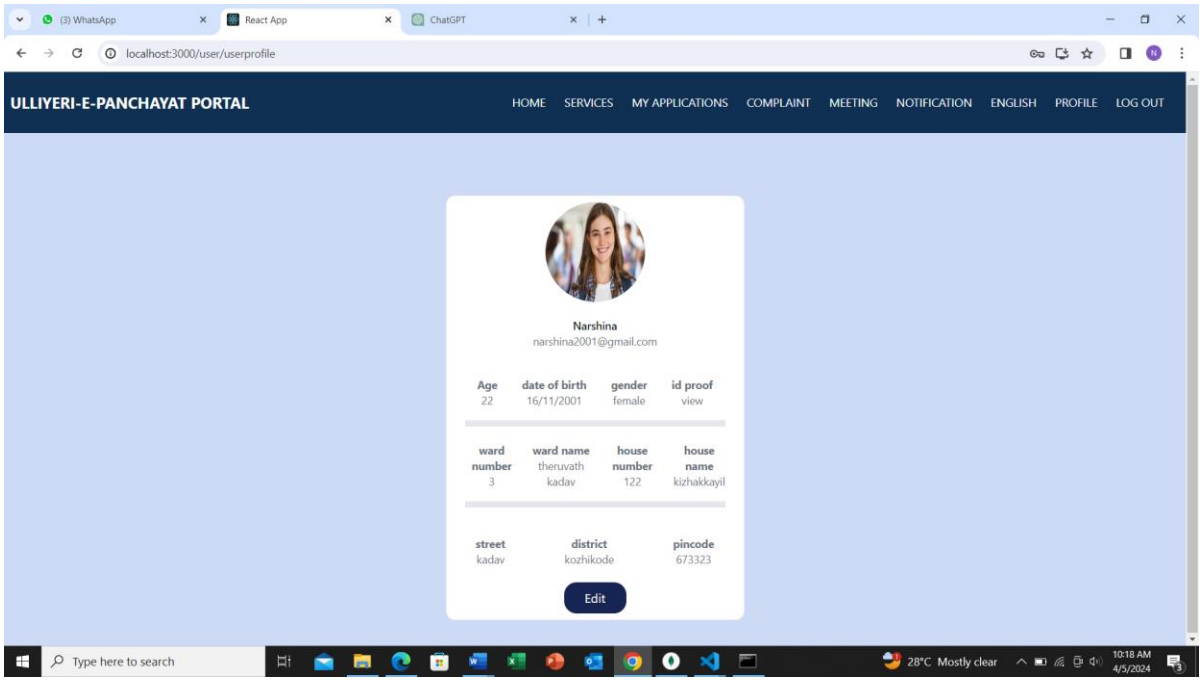
Forgot password



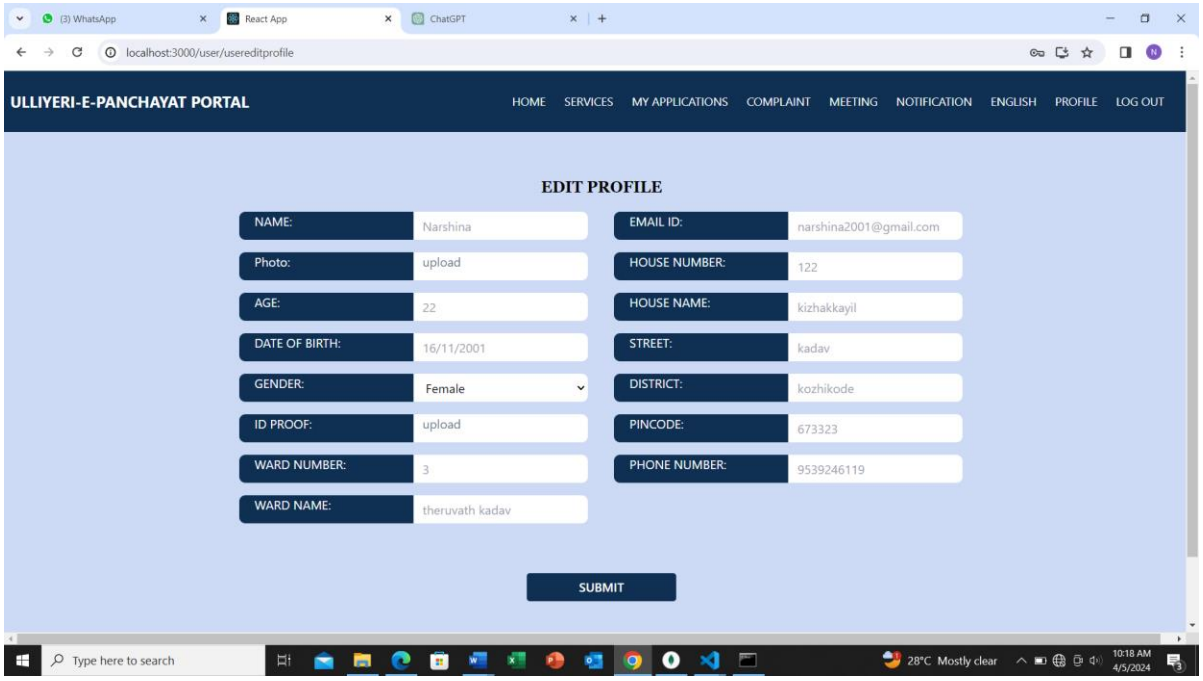
View application status



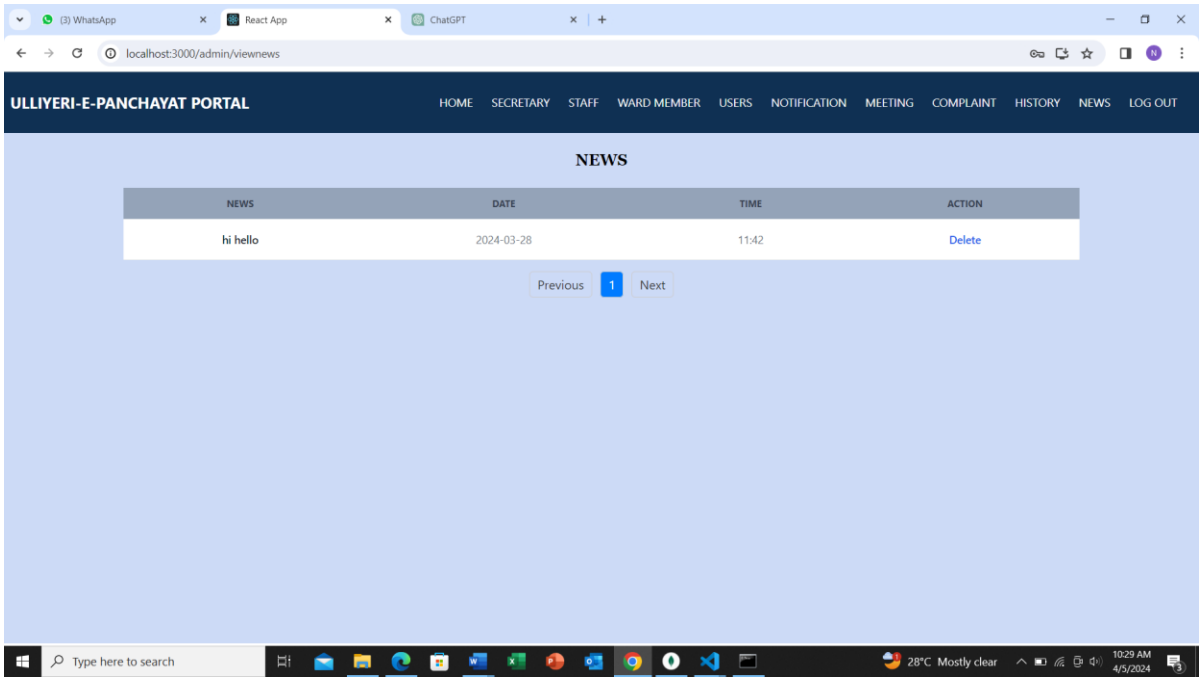
User profile



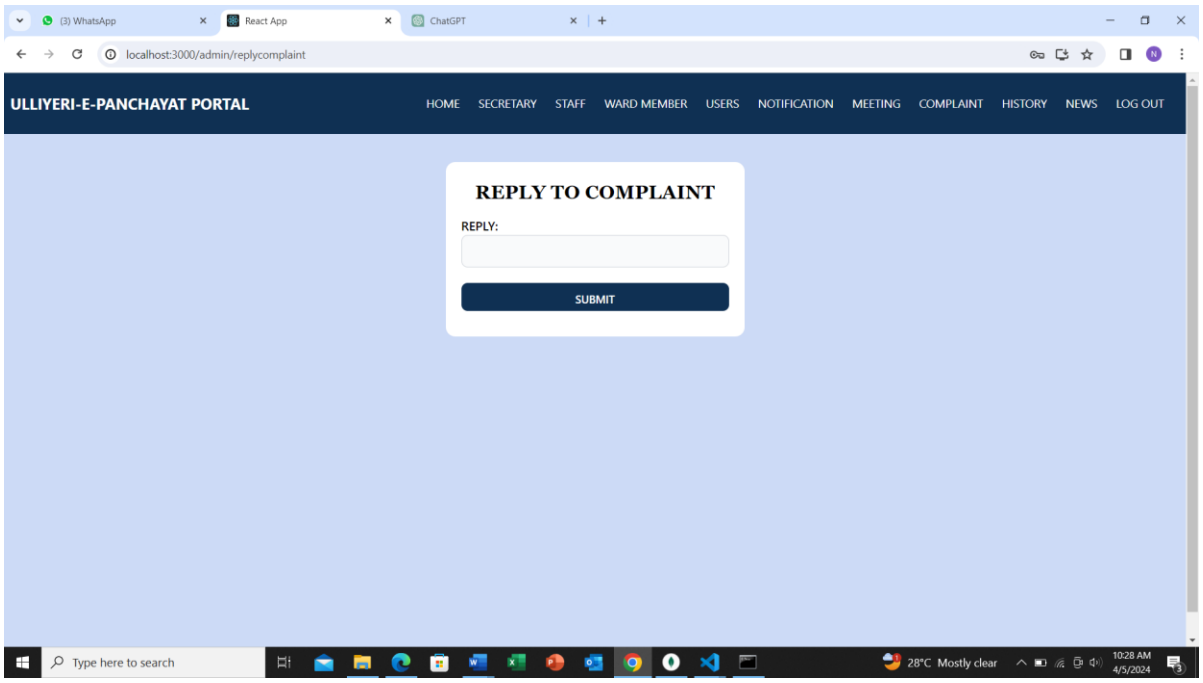
User edit profile



View news



Reply to complaint



BIBLIOGRAPHY

13. BIBLIOGRAPHY

Websites

[1] https://www.mongodb.com/docs (Accessed on date:29/01/2024,3:30pm)

[2] https://expressjs.com (Accessed on date:3/02/2024,7:30pm)

[3] https://nodejs.org/en/docs (Accessed on date:14/02/2024,3:30pm)

[4] https://bootstrapmade.com/onepage-multipurpsode-bootstrap-template(Accessed on date:21/02/2024,3:30pm)

[5] https://developer.mozilla.org/en-US/docs/Web/HTML(Accessed on date:24/02/2024,3:30pm)

[6] https://developer.mozilla.org/en-US/docs/Web/CSS(Accessed on date:12/03/2024,3:30pm)

Books

[1] Thomas A Powell, Fritz Schneider, "JavaScript: The Complete Reference", 3rdEdition, Tata McGraw Hill

[2] Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React,and Node" by Vasan Subramanian

[3] Code complete: a practical handbook of software construction by Steve McConnell, Microsoft Press,2nd Edition (2004).

[4] Full Stack Development with J Hipster: Build scalable and maintainable webapplications using the MERN stack" by Deepu K Sasidharan and Sendil Kumar N

[5] Full Stack JavaScript: Learn Backbone.js, Node.js, and MongoDB" by Azat Mardan