INDIRA GANDHI NATIONAL OPEN UNIVERSITY

MCSP - 060

PROJECT REPORT

E-Gram Panchayath

By

KUMARA V.
(185867320)

Submitted to the School of Computer and Information Sciences, IGNOU in partial fulfilment of the requirements for the award of the degree

Master of Computer Applications (MCA)
2022

Indira Gandhi National Open University
Maidan Garhi
New Delhi - 110068

SCHOOL OF COMPUTER AND INFORMATION SCIENCES
IGNOU, MAIDAN GARHI, NEW DELHI – 110 068

II. PROFORMA FOR THE APPROVAL OF MCA PROJECT PROPOSAL (MCSP-060)

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

Project Proposal No :.....
(for office use only)

Enrolment No.: 185867320
Study Centre: 1314
Regional Centre:..... **RC Code:** 13 Bangalore
E-mail: kumarav013@gmail.com
Mobile/Tel No.: 9620357735

1. Name and Address of the Student: KUMARA V
   #25, Siddhappaji Nilaya, Muddayanapalya, Bangalore-560091
2. Title of the Project***: E-Gram Panchayath
3. Name and Address of the Guide: Vanitha N
   Garuda Nilaya, Kamakshipalya, Bangalore-560079
4. Educational Qualification of the Guide: (Attach bio-data also)

Ph.D*	M.Tech.*	B.E*/B.Tech.*	MCA	M.Sc.*
☐	☐	☐	☑	☐

(*in Computer Science / IT only)

5. Working / Teaching experience of the Guide**: 4 years of Experience in Accenture as Software Engineer
   (***Note: At any given point of time, a guide should not provide guidance for more than 5 MCA students of IGNOU*)
6. Software used in the Project***: Python and Django Webframework, MySQL
   (*** Please refer to section VIII of these guidelines)
7. If already pursued BCA/BIT from IGNOU, mention the title of the project (CS-76) and the s/w used:.....
8. Project title of the Mini Project (MCS-044) and the s/w used: Online Job Portal, s/w : Python and Django
9. Is this your first submission? ☑ Yes ☐ No

Signature of the Student
Date: 25-MAY-2022

Signature of the Guide
Date: 25-MAY-2022

For Office Use Only

☐ Approved ☐ Not Approved

Name:.....

.....
Signature, Designation, Stamp of the Project Proposal Evaluator
Date:.....

Suggestions for reformulating the Project:

Mcasynopsisblr 13 <mcasynopsisblr@gmail.com>
to me

Dear learner,

We have received your synopsis.It is approved.
You may attach this email as the approval letter for the project.

Submit the Hard copy of the project to the Regional Centre in person or by post.

OR

you may upload the final project (if the link provided on the IGNOU Website).

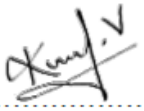
The last date to submit the project is 30th Dec 2021.

Best wishes,
Dr.Hemamalini H.C
Asst.Regional Director

XI. CERTIFICATE OF ORIGINALITY

This is to certify that the project report entitled E Gram Panchayath submitted to **Indira Gandhi National Open University** in partial fulfilment of the requirement for the award of the degree of **MASTER OF COMPUTER APPLICATIONS (MCA)** , is an authentic and original work carried out by Mr. / Ms. Kumara V with enrolment no. 185867320 under my guidance.

The matter embodied in this project is genuine work done by the student and has not been submitted whether to this University or to any other University / Institute for the fulfilment of the requirements of any course of study.



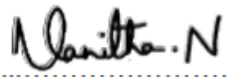
Signature of the Student:

Date: 25-MAY-2022

Name and Address of the student

Kumara V
#25, Siddhappaji niaya
Muddayanapalya,
Bangalore - 560091

Enrolment No. 185867320



Signature of the Guide

Date: 25-MAY-2022

Name, Designation and Address of the Guide:

Vanitha N
Software Enginer
Garuda Nilaya,
Kamkshipalya, Bangalore-79

TABLE OF CONTENTS

sl. No	Topic	Page. No
I.	Introduction & Objective(s)	07
II.	Project Category	08
III.	Tools/Platforms & Hardware SRS	09
IV.	Problem Definition, Project Planning	18
V.	Analysis (DFDs, ER Diagrams etc.)	24
VI.	Implementation of Security Mechanisms	25
VII.	Database	28
VIII.	Coding	30
IX.	Graphical User Interface	36
X.	Testing	44
XI.	Future Scope & Further Enhancement	47
XII.	Bibliography	48

1. Introduction& Objective(s)

Introduction: -

Smart Gram Panchayath is a web application that is specially designed to establish an environment, wherein village public can easily communicate with the village councils.

Our application features: administrator, PDO and village public.

Administrator will handle the whole website and manage the activities of users, PDO will update panchayath services, application, payments and complaints filled by public. The village public will register to the application and avail for the facilities. This post has complete information of Smart Gram Panchayath Project Documentation.

Objective(s): -

- This project is aimed at the developing an E-Gram Panchayat Management System (EGPMS). EGPMS is an online based application that can be accessed throughout the internet. This System may be used for monitoring gram panchayat activities. Admin as well as body member's logging, it may also access and public can search provided information regarding gram panchayat at any time. The Information about Schemes published by government or any other activities and billing record will be updated by body members and the secure data maintained by only administrator. EGPMS is being developed for maintain and facilitate easy access to information. For this user don't need to be registered. It is user friendly website, only admin has authority to give access or choose login members. EGPMS is an online based application that aims to provide information keep the Records and Documents like birth certificates, death certificates, residential certificates, 7/12 certificates. Any user can access the data and download some information from database and only body members have authority to upload the data.
- This application is aimed at achieving the solution for the problems that are faced by the public in villages. Through this application public can easily contact with PDO regarding any issues such as availing for services, complaints filling, bill payments, application filling and feedbacks. The PDO can easily keep track on the village problems and managements.

2. Project Category

This application is developed on using MVT architecture of **Django Web framework** Technologies, **HTML and Bootstrap** is used to front end and **MySQL-Server** which is incorporated with **ORM** (Object Relational Mapper) Technology in Django framework which maps objects with the database which helps to directly access table/database without using MySQL query statements for back end or database. The connection between the front-end and the back-end is established by using Django Database Connector.

3. Tools/Platforms & Hardware/Software Requirement Specification

A. Hardware Requirements: -

Server Configuration (minimum): -

- Intel® Xenon™ 2.0 GHz Processors with 256KB cache.
- 256 MB RAM.
- 200 GB Hard Disk Drive.
- 3.5" – 1.44 MB Diskette Drive.
- 52X CD-ROM Drive.
- Intel® Pro 10/100+ LAN Card.

The server should have a proper backup and recovery facility.

Client Configuration (minimum): -

- Intel® Pentium® 3 1.3 GHz processor with 256 KB Cache.
- 128 MB RAM.
- Intel® Pro 10/100+ LAN Card.
- Connection to the LAN.

The clients should have a good rate of data transfer with the server for quick performance.

B. Software Requirements: -

Server Side:

- Django Web Server
- MySQL-Server
- Python 3.0 or above
- Windows 2003 Server

Client Side:

- Django Web Server
- MySQL-Server
- Python 3.0 or above
- Windows 2003 Server

Characteristics of Back-end…

- Queries against the shared resources
- Management (Application and data)
- Transaction and processing
- Centralized application logic
- Communication and computation

Characteristics of Front-end…

- Pallet of visual tools for designing screens allows the programmers to create sophisticated multiple window user interfaces without writing any code.
- Graphical elements you create on the screen are displayed just as they will appear to the end user.
- The application can be created, designed and run in any web browser.
- The Visual system has a general-purpose programming language.
- Fully supports Event Driven programming.
- Incremental compilation reduces development time.
- Debugging tools provide powerful development support.
- Support for dynamic data Exchange allows interoperability with others Applications.
- Fully supports Object linking and Embedding (OLE).

4. Problem Definition, Requirements Specifications, Project Planning & Scheduling

Problem Definition: -

Now a day's people in the rural areas have to go to panchayat office in their location to apply and get their certificates provided in that office. It requires a lot of time and may result in work delay. The data in the office has to be maintained manually. There is no security for the data and faults can be encountered during entering the data mainly which require higher calculations. People also face so many problems in their area. They complain to their respective ward members but they may or may not respond quickly. There are many other problems in the present-day panchayat raj system.

This phase consists of…

- System Engineering
- Software Requirement Analysis
- Software Design

Problem Recognition…

In this phase the analyst should clarify –

- Who is behind the request for this work?
- Who will use the solution?
- What will be the economic benefit of the successful solution?
- Is there any other source for the solution that the organization needs?

Getting answers all those questions by himself/herself, the analyst starts working and he issues some questionnaires to the future users for gathering information and then after those concentrates on the feasibility studies.

Preliminary Investigations…

The purpose of the preliminary investigation is to collect information for developing broad solutions for the purpose of feasibility study. Material i.e., information and facts to be collected in the preliminary investigation not only act as a basis for forming the several board solutions of proposed system but it also provides the much-needed feedback for selection of the final candidate's system among the solutions suggested in the course of feasibility study. Actually, it means to be found out the way that how the proposed site will

be developed containing which facts and figures. Following the four broad methodologies, which are described as follows: -

- Interview.
- Questionnaire.
- Fact finding studies, etc.

After the verbal interview session is over, it was very clear that the team requires to Requirements Specifications.

Requirements Specifications: -

In the traditional system files were used to maintain the database which was done manually. This existing system consumes a lot of time. This time-consuming evaluation coupled by the huge maintenance problem and may also lead to erroneous results. The various operations performed on these files like sorting, adding, modifying and deletion of the records are very tedious. Moreover, these manually maintained files have the possibility of getting worn out. Thus, less durability is achieved.

Thus, the demerits of the existing system can be summarized as follows…

- There is no consistency as the data may not be updated on time.
- Feasibility is reduced
- Less reliability
- Security is not provided and any one can access
- Prioritization of records is difficult.
- More erroneous
- Difficult to maintain
- As everything is done manually its slow process
- No timely acknowledgement services

Taking the demerits into consideration, an alternative system which uses Oracle as both front end and back end was used. In front end, retrieval of the data from the database is done through SQL queries i.e., using D2K forms. This is not a web application and the data is not distributed as only a single system is used. As it is confined only to a particular system, scope is limited and there is a hindrance to the reliability if the system fails.

The demerits of this alternate system are…

- Only single system used.
- If the system crashes, then the data is lost
- too overburdened

- not reliable
- slow processing
- less flexible
- not so user friendly

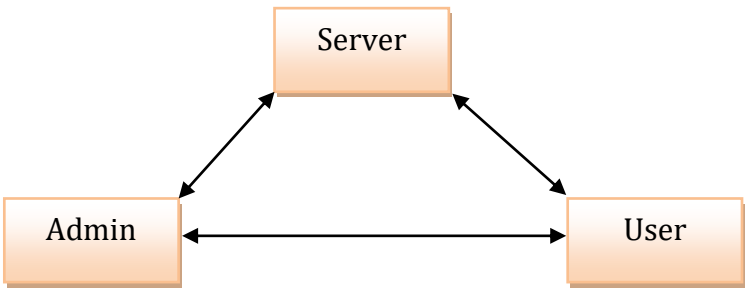
Proposed system: -

E-Gram Panchayat Management System (EGPMS) propose solutions to all the problems in the current system. EGPMS provides online service to the people living in that area. All the services which are done manually are made online in the project. The people can about their panchayat, activities notifications and all other information related their villages. All the applications and certificates are applied and verified online. The users on the people in the village can complain about their problem through online. Suggestions are also accepted from the people for the development of their village. The user can request any application, suggestion, and complaint at anywhere and at any time. The gram panchayat provides birth certificate, death certificate, residential certificates, 7/12 certificates, domicile certificate, receipts for house tax, water tax etc. They give order for construction of road, buildings, renewal of building. They keep records of their monthly & yearly budge.

The objectives of the proposed system are as follows:

- Easy to use, effective and efficient
- Accurate results.
- Easy maintenance.
- Fast access
- More feasibility
- More secure.
- Provides high consistency.
- More reliable

System Architecture: -



Feasibility Study…

Feasibilities are studied from Economical, technical, operational and legal point of view and hence found no obstacles to continue with our proposed project to be developed. So, the feasibility studies are undergone as follows:

- **Economic Feasibility:** More commonly known as Cost/Benefit Analysis. The procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If the benefits outweigh costs, then decision is made to design and implement the system. Considering the facts, it is becoming evident that the system will be economically feasible both for developer as well as for client's respect.
- **Technical Feasibility:** Technical feasibility centers on the existing computer system (hardware, software, etc.) and to what extent it can support the proposed addition. If the budget is a serious constraint, then the project is judged not feasible. In our case this does not become an obstacle.
- **Behavioral Feasibility:** People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. This was not such a problem in our mentioned project.
- **Legal Feasibility:** A determination of any infringement, violation or liability that could result from the development of the system. But the system to be developed will be 100% legal.
- **Operational Feasibility:** The management & operators desire to be well acquainted with the requisite skill needed. Here most of the members in development team having technical expertise.
- **Time Feasibility:** The management & operators here concern about whether the project will be completed timely or not. But considering the facts and figures collected by us regarding our project it can be easily assumed that the project will be completed within the specified time frame

Cost and Benefit Analysis…

In developing the cost estimates for this system several cost elements were taken into consideration. Among them was hardware, software, facility, operating and supply costs. Hardware costs related to the actual purchase of the computer and peripherals (for example, printer, disk drive, tape unit). Determining the actual cost of hardware is generally more difficult when various users than for dedicated stand-alone system share

the system. Software costs relate to the buying of the software required to develop the project as well as the software required to run the application in the organization.

Functional Requirements: -

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that show how a use case is to be fulfilled. They are supported by non-functional requirements, which impose constraints on the design or implementation (such as performance requirements, security, or reliability).As defined in requirements engineering, functional requirements specify particular behaviors of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability.

Technical Requirements: -

- Performance Requirements
- Safety Requirements
- Security Requirements
- Hardware Constraints
- Software Constraints
- Design Constraints

Nonfunctional Requirements: -

In systems engineering and requirements engineering, non-functional requirements are requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that specify specific behavior or functions. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals" and "quality of service requirements". Qualities, of Non-functional requirements can be divided into two main categories. Execution qualities, such as security and usability, are observable at run time. Evolution qualities, such as extensibility and scalability, embody in the static structure of the software system.

Software Quality Attribute-

- **Reliability:** Reliability will mostly depend on the client's connection status. The reliability holds as long as clients are provided support to JDK 1.5 or later.
- **Availability:** The server on which this system will be running is expected to be available at all hours of the day to provide worldwide accessibility. A graphical user interface (GUI) is provided to have online interaction with the user.
- **Maintainability:** The business logic here is that there can be accounting feature in the future development of the product. All development will be provided with good documentation.
- **Portability:** As the system is designed using Java, it can work on any platform or architecture

Project Planning & Scheduling: -

PERT Chart/ Task Network Chart: -

The Program Evaluation Review Technique (PERT) is the cost and time management system. PERT organizes that project is complex that some tasks must be completed before other can be stated and that the appropriate way to manage a project is to define and control each task. Because projects often fall behind schedules, PERT is designed to facilitate getting a project back on schedule. The PERT chart gives a graphical representation of this information.

Depending on the working priorities, the entire project can be subdivided into the following main modules, those are: -

- User Module
- Administrator Module
- PDO Module
- Service/Complaint Module
- Bill Payment Modules
- Certificates Module

.

We can construct our activities plain as follows: -

Activity	**Activity Name**
A	Login
B	Administrator
C	PDO
D	Service/Complaint
E	Bill Payment
F	Certificates

Chart: -

Activity	Predecessor Activity	Time Estimated Weeks (Individual)
A	.....	3
B	A	4
C	B	4
D, E	C	5
F	D, E	3

Critical Path Method (CPM): -

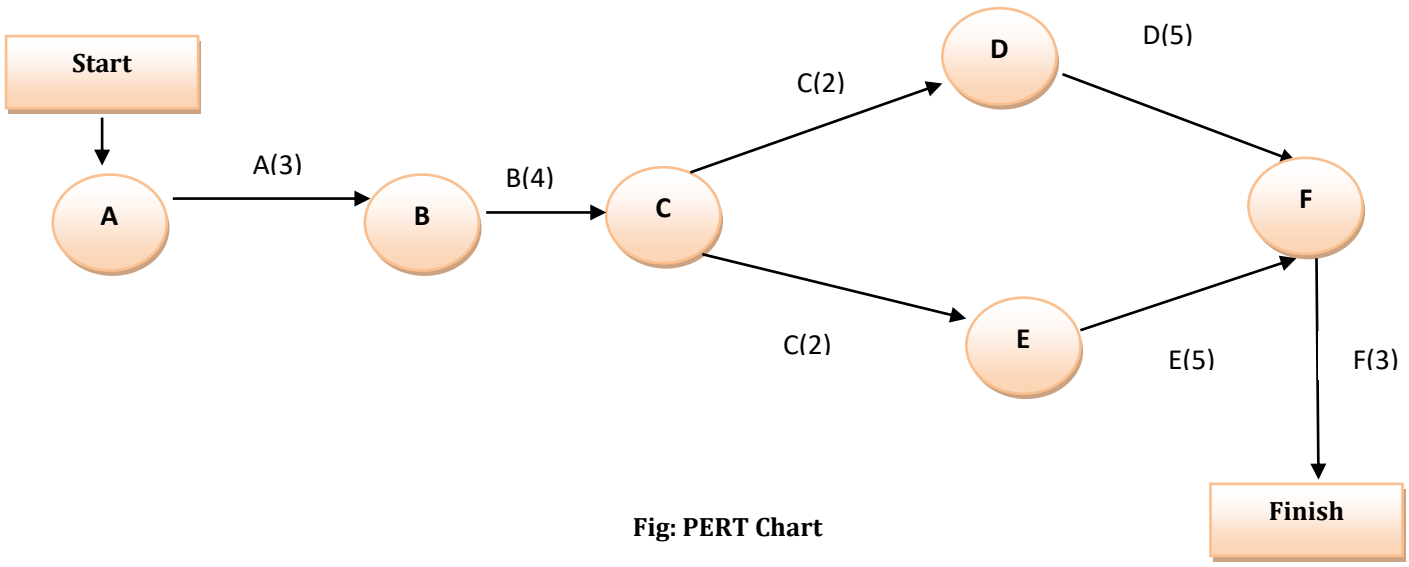


Fig: PERT Chart

Time Line Path: -

Path	Length Of Time
Start-A-B-C-D-F-Finish	4+4+2+5+2= 18WEEKS
Start-A-B-C-E-F-Finish	4+4+2+5+2= 18 WEEKS

GANTT Chart: -

A Gantt chart is a popular type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project.

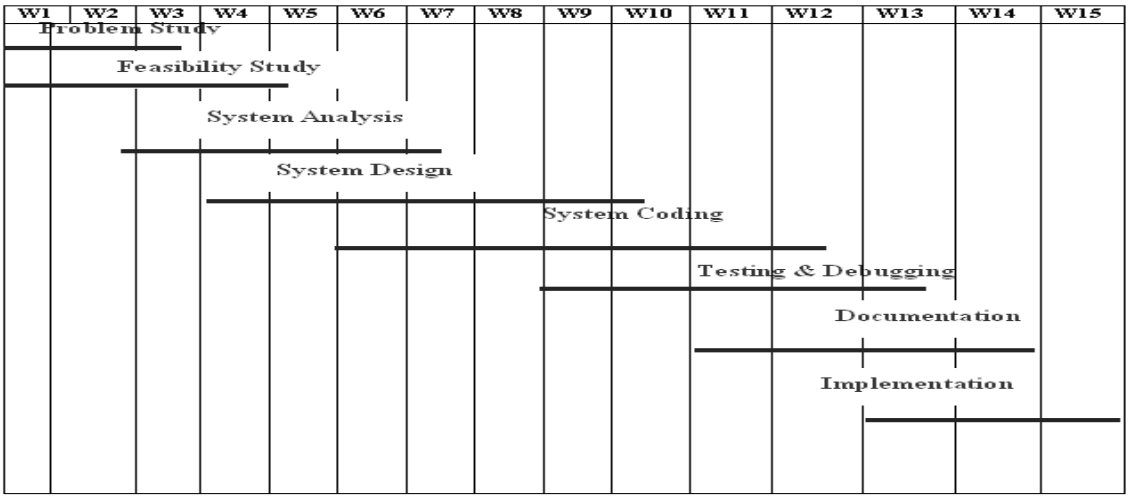


Fig: GANTT Chart

5. Analysis (DFDs, ER Diagrams etc.)

The whole approach of analysis of problem should however be based around critical factors like the availability of information for making the decision, the time available for processing the data i.e., the realism. System Requirement Specification or SRS had been prepared after proper discussion with the persons attached with the mentioned "**OSCM**". Software project management begins with a set of activities collectively called PROJECT PLANNING. Software project planning actually encompasses all of the activities. Planning involves estimation- to determine how much money, how much effort, how many resources, and how much time it will take to build a specific software-based system or product.

Phases Cover: -

- Pre–Analysis Studies
- System Analysis
- System Design
- Project Coding
- Project Testing
- Implementation & Documentation

- **Pre–Analysis Phase:** In this phase problems with existing system are to be determined and do the investigation to make the solutions.
- **System Analysis Phase:** In this phase system analysis is done by preparation of Software Requirement Specification.
- **System Design Phase:** The purpose of the design phase is to plan a solution for the problem specified in the requirements documents.
- **Project Coding Phase:** The goal of the coding phase is to translate the design of the system into a program code by help of a programming language like Visual Studio, Java, etc.
- **Project Testing Phase:** Testing concerned with the elimination of errors introduced during coding phase.
- **Implementation & Documentation Phase: -** This phase includes all the activities performed to keep the system operational after the installation of the software.

Data Flow Diagram (DFDs): -

Data flow diagrams models the passage of data in the system and are represented by lines joining system components. Flows of data in the system can take place between: -

- Between two processes,
- From a data store to a process,
- From a process to data store,
- From a source to process and
- From a process to a sink.

Though the system mainly consists of two parts viz. online admission and online examination and other parts are going to be automated gradually…, so DFD is also, illustrated in two parts, respectively…

Context Level Diagram: -

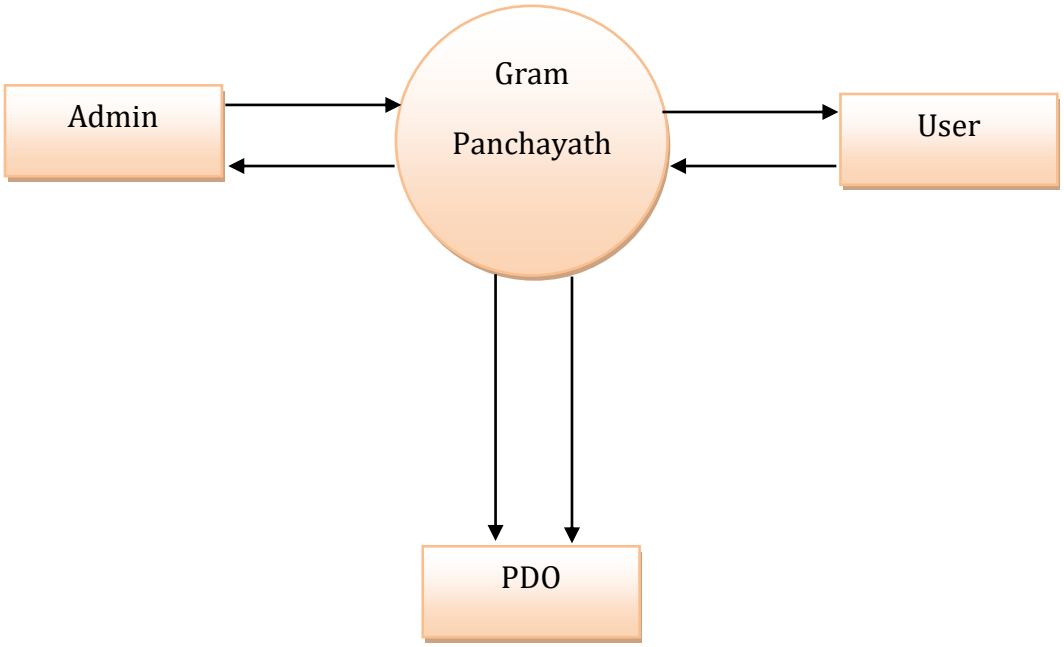


Fig: Context Level Data Flow Diagram

PDO Level 1 Diagram: -

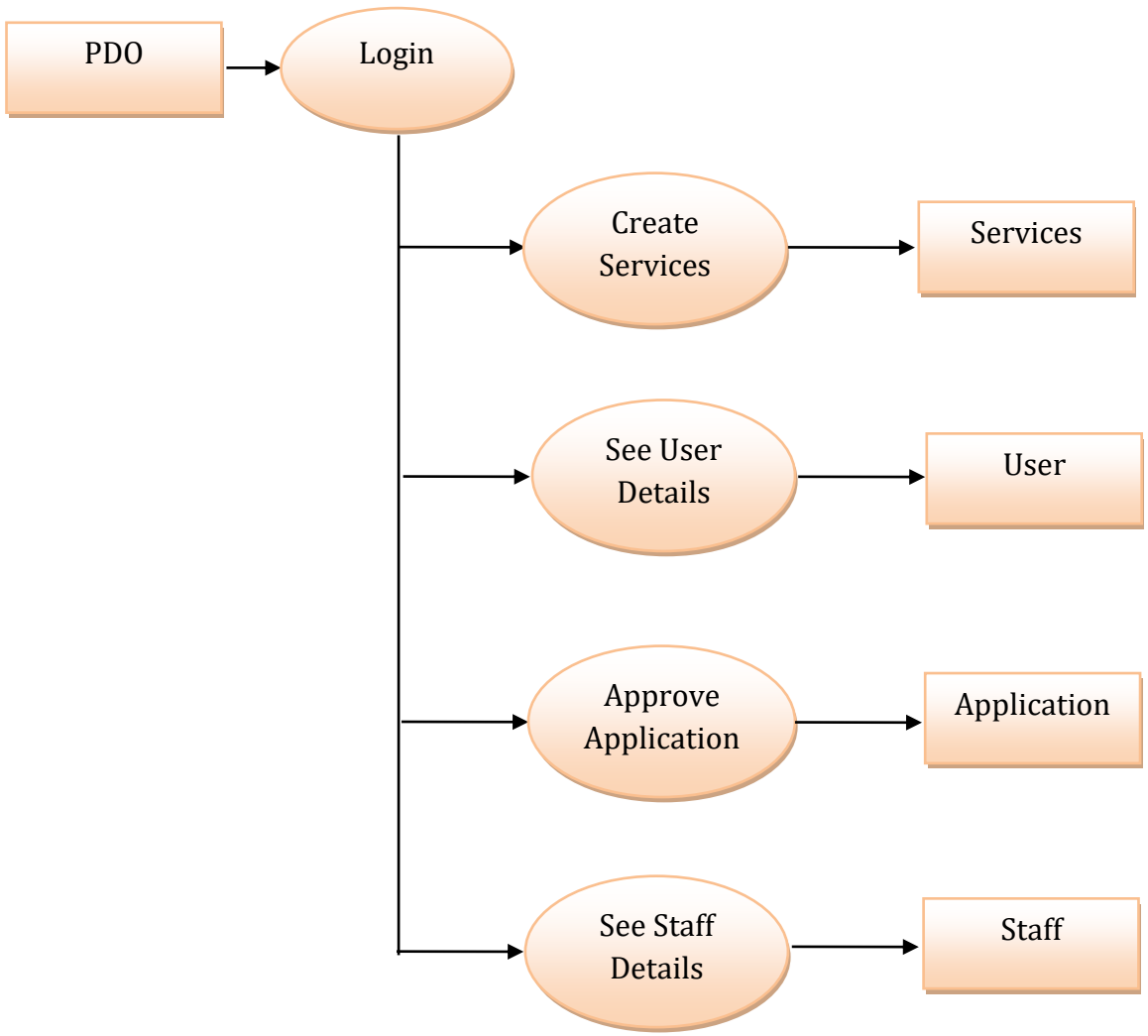


Fig: PDO Data Flow Diagram

User Level 1 Diagram: -

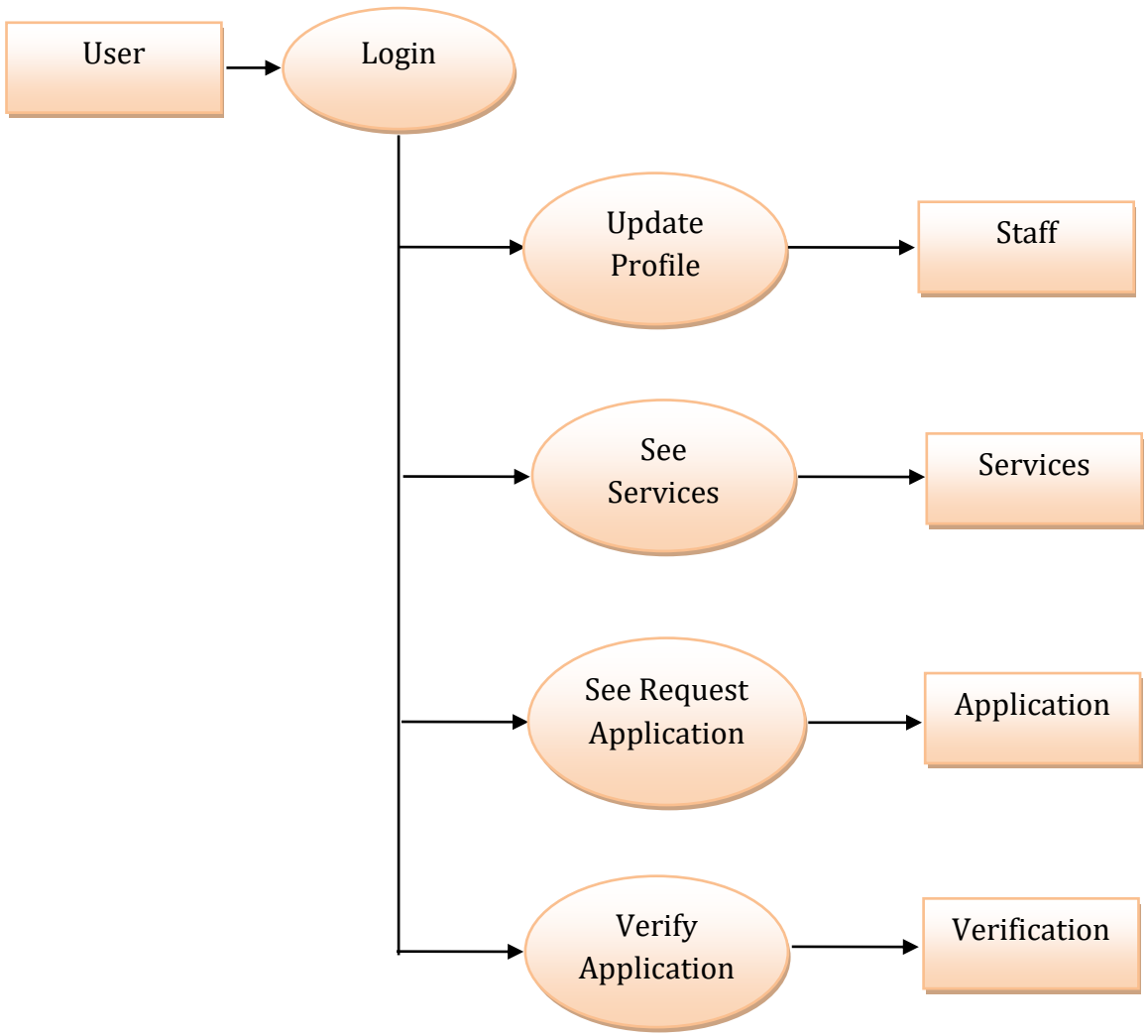
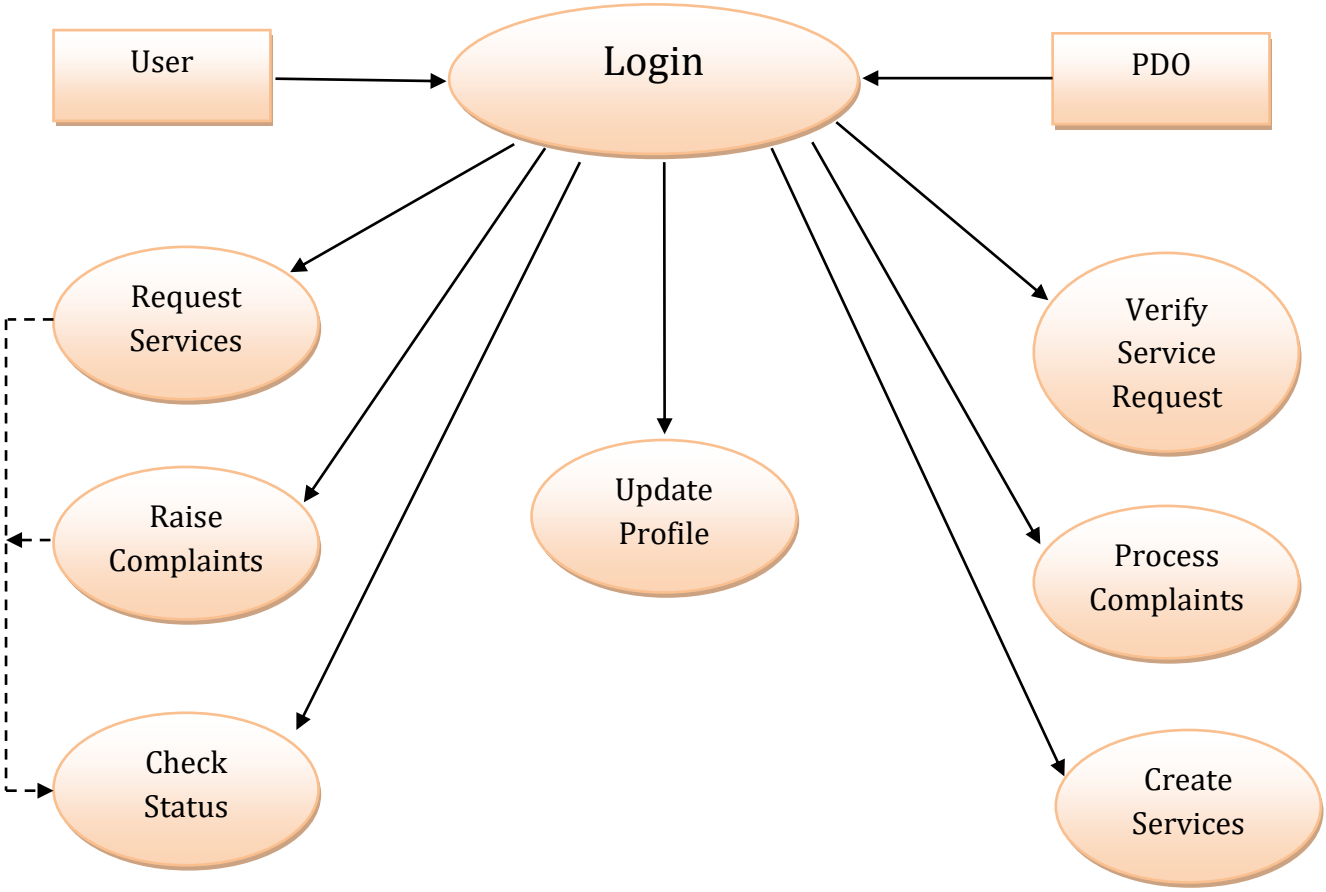
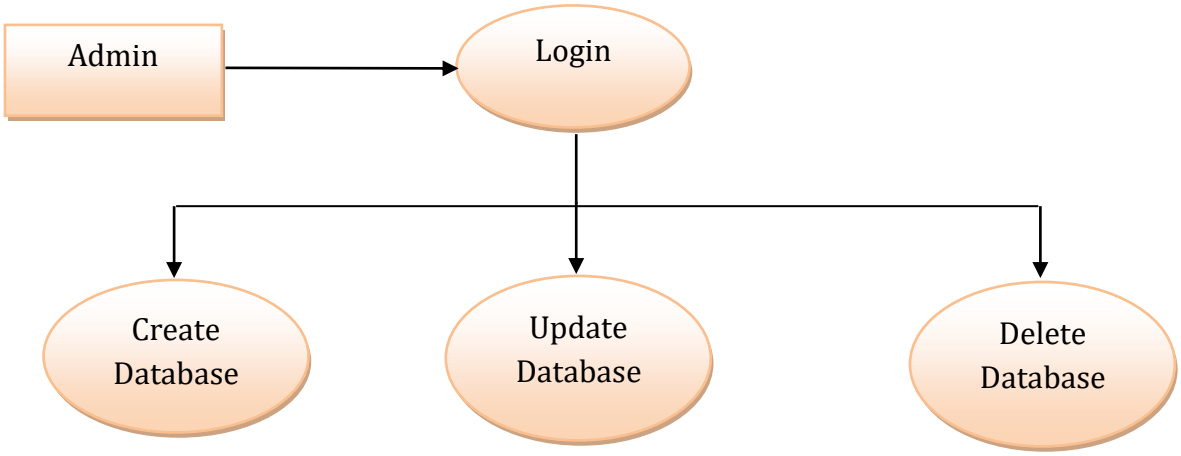


Fig: User Data Flow Diagram

Level 2 Diagram: -



Admin Data Flow Diagram: -



Entity Relationship Diagram (ERD): -

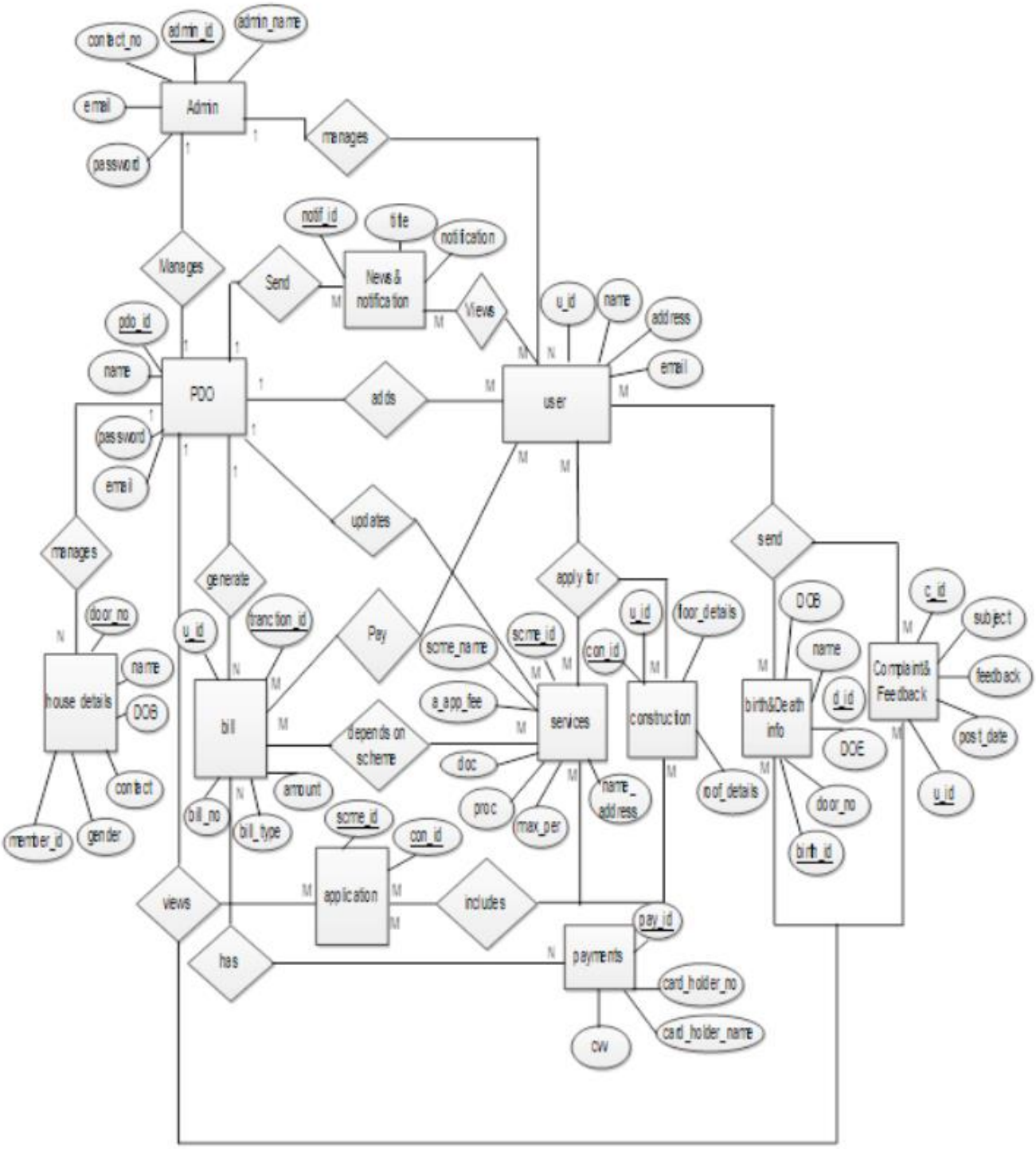


Fig: E-Gram Panchayath Entity Relationship Diagram

6. Django Architecture

Django MVT

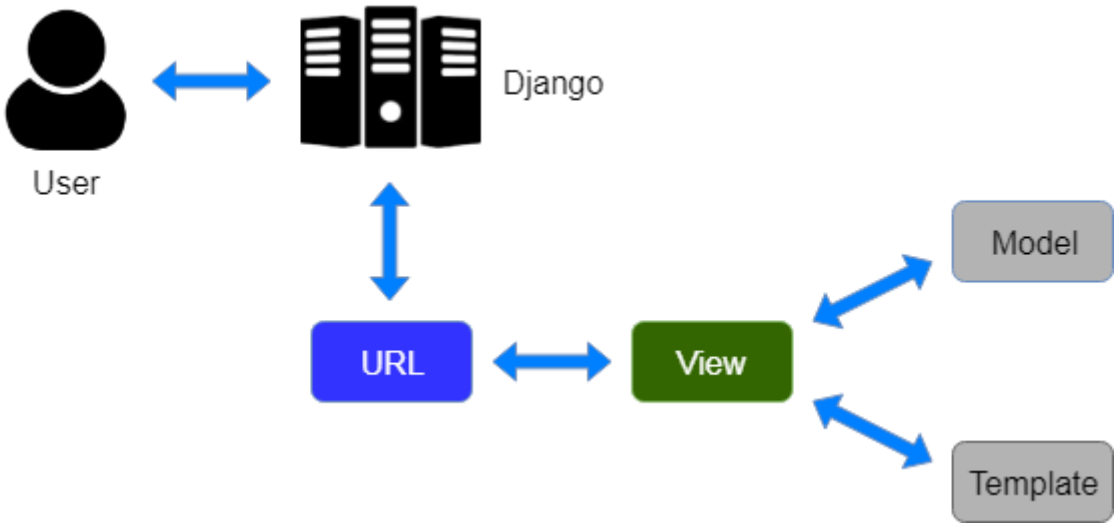
The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data.

The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template.

Although Django follows MVC pattern but maintains its own conventions. So, control is handled by the framework itself.

There is no separate controller and complete application is based on Model View and Template. That's why it is called MVT application.

See the following graph that shows the MVT based control flow.



Here, a user **requests** for a resource to the Django, Django works as a controller and check to the available resource in URL.

If URL maps, **a view is called** that interact with model and template, it renders a template.

Django responds back to the user and sends a template as a **response**.

7. Implementation of Security Mechanisms at Various Levels

Regarding security a project it is very important matter to having proper protection mechanism from unauthorized users. So, here is also introduced some of the security measures which will be enhanced much in later versions. Some of the security measures are as follows: -

Entry-level password and user name for both administrator and general users: -

- Try to prevent the entry of an unauthenticated user.
- More than three times attempt to login as an intruder is prohibited.
- Administrator's level security again applicable for bill generation.

Those are some of the important measures for security maintenance of this Hospital management package, which is also having LAN facilities, too.

A. System Security: -

a. Access Control:

Individual users shall be allowed access into the system by including their names, authorized points of access and log-in ID in the user database. This table shall be accessible to the system administrator only.

Controlling access to the computer facility is secured through the following steps: -

- The system needs administrative privilege to operate through round the clock.
- The software supports/enforces that there will be very strict checking in case of user id & password. The system is totally password protected that no unauthorized user may get any information about the system.

b. Authorization:

Authorization is very important for software. Authorization is the first step of software entrance means without valid user id and password no one can get information about this project.

c. Integrity:

Integrity constraints implemented here ensure that any properly authorized access; alteration network node IP address doesn't change the consistency and validity of the configuration and discovery.

d. Identification:

It is a scheme for identifying the valid user of the system through the following mechanisms:

- Authentication of user Id and password.
- Validation of user name

B. Database Security: -

- In case of database each user needs to gain access to MySQL Server through a login account that establishes the ability to connect (authentication).
- This login then has to be mapped to a MySQL Server user account used to control activities performed in the database (permissions validation).

Besides the security measures inherent to the OS and the database manager, the following additional security measures will be adopted: -

C. Hierarchical Security: -

The security environment in MySQL Server is stored, managed, and enforced through a hierarchical system of users. To simplify the administration of many users, SQL Server uses groups and roles: -

- A group is an administrative unit within Microsoft Windows NT 4.0 and Windows 2000 that contains Windows NT 4.0 and Windows 2000 users or other groups.
- A role is an administrative unit within SQL Server that contains SQL Server logins, Windows NT 4.0 and Windows 2000 logins, groups, or other roles.

Arranging users into groups and roles makes it easier to grant or deny permissions to many users at once. The security settings defined for groups are applied to all members of that group. When a group is a member of a higher-level group, all members of the group inherit the security settings of the higher-level group, in addition to the security settings defined for the group itself or user accounts.

The requirements of a database security system go beyond this one-manager limitation; employees belong to security groups that do not fall within the strict organizational plan of the company. For example, administrative staff exists in every branch of the company and require security permissions regardless of their organizational branch. To support this broader model, the security system in Windows NT 4.0, Windows 2000, and MySQL Server allows groups to be defined across a hierarchy. This hierarchical system of security groups simplifies management of security settings. It allows security settings to be applied collectively to all group members, without having to be defined redundantly for each

person. The hierarchical model also accommodates security settings applied only to a single user.

D. System Failures and Recovery: -

System security is closely linked with system failures and recovery. Possible failures and recovery mechanisms are incorporated within this system are mentioned as follows:

- **Catastrophic Failures:**

  Are restored using the roll forward method of recovery.

- **Logical Failures:**

  The system developed is an interactive system and it provides automatic recovery.

- **Structural Failure:**

  If this fault occurs then the systems will automatically the log file is stored up till how much it is captured.

- **Consistency Error:**

  The system includes routines that check the consistency of information entered in the configuration and possessed discovery.

E. Data Security: -

One of the functions of a database is to protect the data by preventing certain users from seeing or changing highly sensitive data and preventing all users from making costly mistakes. The security system in MySQL Server 7 controls user- access to the data, and user-permissions to perform activities.

8. DATABASE

TABLE NAME: USERS

Column Name	Column Type	Condition
User Name	Char Field	Max length = 150
First Name	Char Field	Max length = 150
Email	Email	
Password	Char Field	Max length = 40
Groups	Char Field	
User Permission	Char Field	
Is Staff	Boolean Field	Null = True
Is Active	Boolean Field	Null = True
Is Super User	Boolean Field	Null = True
Last Login	Date Field	Time Zone
Date Joined	Date Time Field	Default = time zone

TABLE NAME: PROFILE

Column Name	Column Type	Condition
User	One to One Field	On delete User = Delete Profile
Profile Pic	Image Field	Default profile
Birth Date	Date Field	
Phone No.	Char Field	Max length = 10
Address	Char Field	Max length = 200
Qualification	Char Field	Max length = 100
Department	Char Field	Max length = 100

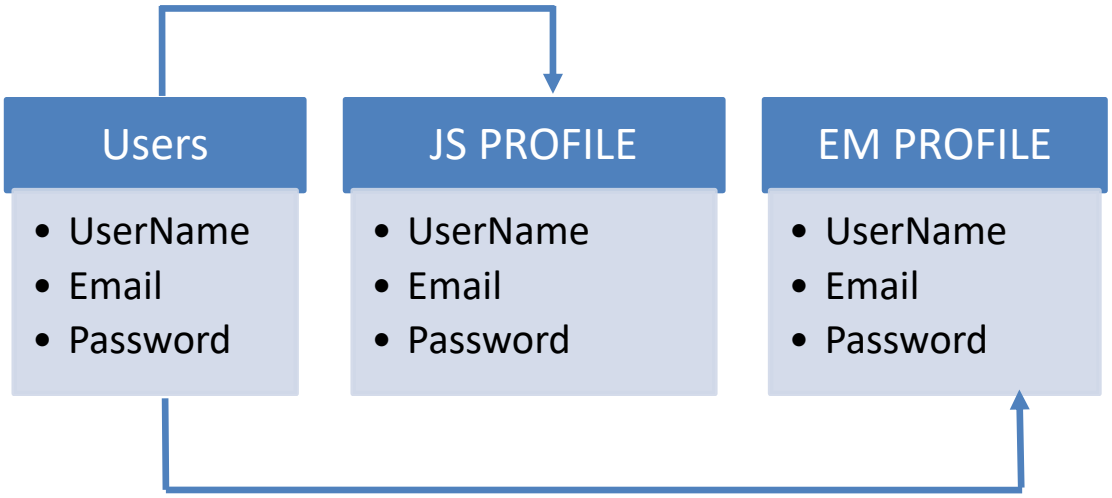
TABLE NAME: SERVICES

Column Name	Column Type	Condition
Service Type	Char Field	Max length = 20
Service Name	Char Field	Default profile
Service Applied Date	Date Field	
Service Description	Text Field	
Service Applied By	One to One field	On delete User = Delete Service
Serv Request Image	Image Field	
Serv Remarks	Text Field	Max length = 100
Estimated Completion Date	Char Field	Max length = 20
Completion Status	Char Field	Max length = 20
Serv Completion Image	Image Field	
Serv Doc Name	Char Field	
Serv Doc File	File Field	

TABLE NAME: CERTIFICATES

Column Name	Column Type	Condition
Certificate Type	Char Field	Max length = 20
Certificate Name	Char Field	
Father Name	Char Field	Max length = 20
Mother Name	Char Field	Max length = 20
Address	Text Field	
Govt id	Char Field	Max length =20
Govt Proof Document	File Field	
Birth date	Date Field	
Death Date	Date Field	
Death Location	Char Field	Max length = 20
Cast	Char Field	Max length = 20
Cast Category	Char Field	
Income Value	Char Field	
Document Proof	File Field	
Remarks	Text Field	
Approval Status	Char Field	Max length = 20

TABLE CONNECTIVITY:



9. CODING

Login Function:

```
def login_user(request):
        if request.method == 'POST':
                uname = request.POST.get('uname')
                pwd = request.POST.get('pwd')
                user = authenticate (username=uname, password=pwd)
       if user is not None:
              login(request, user)
              return redirect('Posts')
       else:
              messages.warning(request, "Username and Password is Invalid")
              return render (request, 'Users/Login.html')
```

Register Function:

```
if ph.exists():
        messages.warning(request, f'Phone number already exists, please provide your
        personal phone number')
elif mail.exists():
        messages.warning(request, f'Email ID already exists, please provide your personal
        Email address')
elif not (ph.exists() and mail.exists()):
        userdetail = User.objects.create(username=email, first_name=fname,
        last_name=lname, email=email)
        userdetail.set_password(password)
        userdetail.save()
        userprofile = JSProfile.objects.create(user=userdetail, BirthDate=birthday,
        phoneno=cnumber, address=address, Resume=myfile)
        userprofile.save()
        send_mail('Registratoin', 'Thank you for registering to Job portal(Trial) Version',
        settings.EMAIL_HOST_USER, [email], fail_silently=False, )
        messages.success(request,
                'Account created successfully for "{}" Please login and complete your
                profile'.format(email))
        return redirect('Login')
```

Password Change Function:

```
 if request.method == 'POST':
      form = UserPwdForm(request.user, request.POST)
      if form.is_valid():
         form.save()
         messages.DEBUG(request, f'Your password has successfully updated, please login.')
         return redirect('Login')

 else:
      form = UserPwdForm(request.user)
```

My Request Class

```
class MyRequests(ListView):
   model = Services
   template_name = "Services/MyRequests.html"
   context_object_name = 'requests'
   paginate_by = 5

   def get_queryset(self):
      req = Services.objects.filter(Serv_Applied_By=self.request.user).order_by('-
Serv_Applied_Date')

      def ExecQuery(req, Serv):
         key = list(Serv.keys())[0]
         if key=='Serv_Type':
            req = req.filter(Serv_Type=Serv[key]).order_by('-Serv_Applied_Date')
         if key=='Serv_Approval_Status':
            req = req.filter(Serv_Approval_Status=Serv[key]).order_by('-Serv_Applied_Date')
         if key=='Serv_Completion_Status':
            req = req.filter(Serv_Completion_Status=Serv[key]).order_by('-Serv_Applied_Date')
         return req

      if self.request.GET.get('idServType'):
         req = ExecQuery(req, {'Serv_Type':self.request.GET.get('idServType')})

      if self.request.GET.get('idAppStatus'):
         req = ExecQuery(req, {'Serv_Approval_Status':self.request.GET.get('idAppStatus')})

      if self.request.GET.get('idCompStatus'):
         req = ExecQuery(req,
{'Serv_Completion_Status':self.request.GET.get('idCompStatus')})

      return req
```

Received Service Requests Class:

```
class ReceivedRequests(ListView):
    model = Services
    template_name = "Services/RequestsReceived.html"
    context_object_name = 'requests'
    paginate_by = 5

    def get_queryset(self):
        req = Services.objects.all()

        def ExecQuery(req, Serv):
            key = list(Serv.keys())[0]
            if key=='Serv_Type':
                req = req.filter(Serv_Type=Serv[key])
            if key=='Serv_Approval_Status':
                req = req.filter(Serv_Approval_Status=Serv[key])
            if key=='Serv_Completion_Status':
                req = req.filter(Serv_Completion_Status=Serv[key])
            return req

        if self.request.GET.get('idServType'):
            req = ExecQuery(req, {'Serv_Type':self.request.GET.get('idServType')})

        if self.request.GET.get('idAppStatus'):
            req = ExecQuery(req, {'Serv_Approval_Status':self.request.GET.get('idAppStatus')})

        if self.request.GET.get('idCompStatus'):
            req = ExecQuery(req,
{'Serv_Completion_Status':self.request.GET.get('idCompStatus')})

        return req
```

Process Request Class View:

```
class ProcessRequest(DetailView):
    model = Services
    template_name_suffix = '_process_request_details'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)

        return context
```

About Us View Function:

```
def About(request):
   return render(request, 'Services/about_us.html')
```

Service Request Code:

```
def ServiceRequest(request):
   if request.method == 'POST':
      requested_by = request.user
      serv_name = request.POST.get('ReqName')
      serv_type = 'Service'
      doc_name = request.POST.get('DocName')
      state = request.POST.get('state_name')
      city = request.POST.get('city_name')
      location = request.POST.get('Location')
      applied_date = datetime.datetime.now()
      desc = request.POST.get('ReqDesc')

      if request.FILES.__contains__('file'):
         doc_proof = request.FILES['file']
         fs = FileSystemStorage()
         fs.save(doc_proof.name, doc_proof)


      if location:
         service_data = Services.objects.filter(Serv_State=state, Serv_City=city,
Serv_Type=serv_type,
                                    Serv_Name=serv_name, Serv_Applied_By=requested_by,
                                    Serv_Location=location)
      else:
         service_data = Services.objects.filter(Serv_State=state, Serv_City=city,
Serv_Type=serv_type,
                                    Serv_Name=serv_name)
      if service_data.exists():
         messages.warning(request, f'Request Already Raised, please find the Request Details
here')
      else:
         if location:
            serv_details = Services.objects.create(Serv_Type=serv_type,
Serv_Name=serv_name, Serv_State=state,
                                       Serv_City=city, Serv_Applied_By=requested_by,
                                       Serv_Location=location, Serv_Approval_Status='pending',
                                       Serv_Completion_Status='NA',
Serv_Applied_Date=applied_date,
                                       Serv_Doc_Name=doc_name, Serv_Doc_Proof=doc_proof,
                                       Serv_Description=desc)
```

```
        else:
            serv_details = Services.objects.create(Serv_Type=serv_type,
Serv_Name=serv_name, Serv_State=state,
                                    Serv_City=city, Serv_Applied_By=requested_by,
                                    Serv_Approval_Status='pending',
                                    Serv_Completion_Status='NA',
Serv_Applied_Date=applied_date,
                                    Serv_Doc_Name=doc_name, Serv_Doc_Proof=doc_proof,
                                    Serv_Description=desc)
        serv_details.save()
        messages.warning(request,'Service Request Submitted Successfully')


    return render(request, 'Services/Services.html')
```

Service Complaint Request Function:

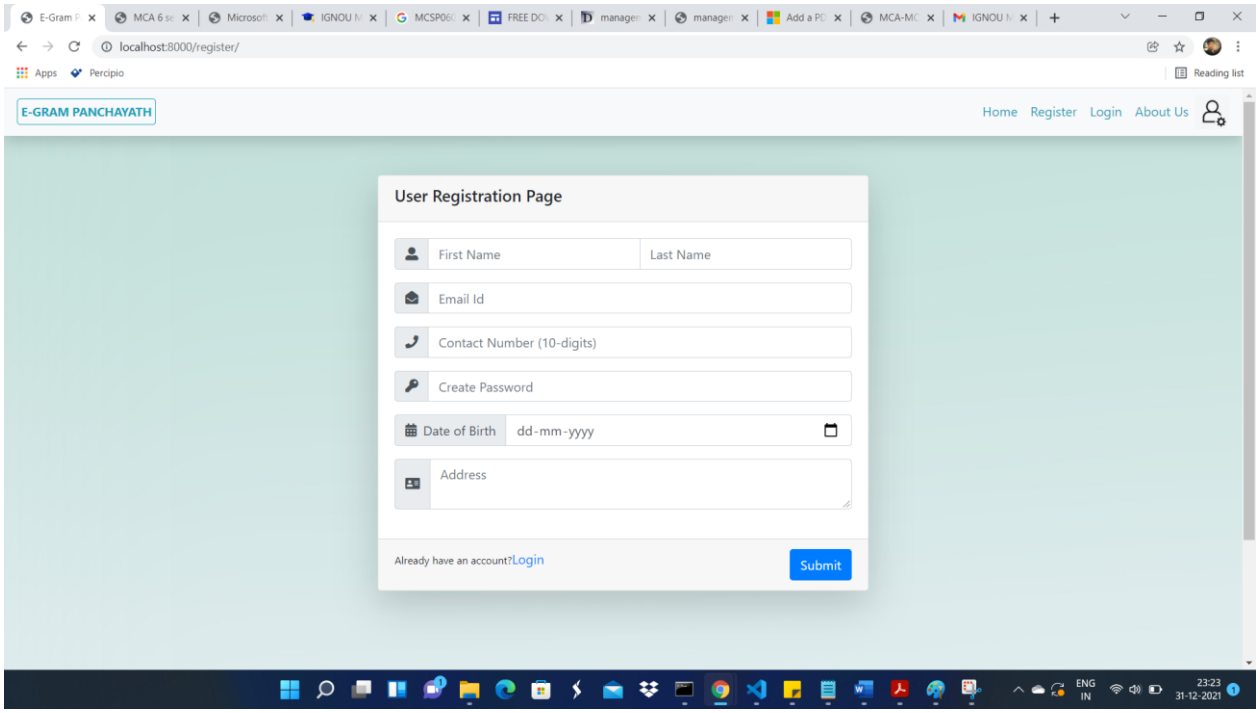
```
def ServiceComplaint(request):
    if request.method == 'POST':
        requested_by = request.user
        serv_name = request.POST.get('ReqName')
        serv_type = 'Complaint'
        doc_name = request.POST.get('DocName')
        state = request.POST.get('state_name')
        city = request.POST.get('city_name')
        location = request.POST.get('Location')
        applied_date = datetime.datetime.now()
        desc = request.POST.get('ReqDesc')

        if request.FILES.__contains__('CImage'):
            Image = request.FILES['CImage']

            fs = FileSystemStorage()
            fs.save(Image.name, Image)

        if request.FILES.__contains__('doc_proof'):
            doc_proof = request.FILES['doc_proof']
            fs = FileSystemStorage()
            fs.save(doc_proof.name, doc_proof)


        service_data = Services.objects.filter(Serv_State=state, Serv_City=city,
Serv_Type=serv_type,
                                    Serv_Name=serv_name, Serv_Applied_By=requested_by,
                                    Serv_Location=location)

        if service_data.exists():
```

```
        messages.warning(request, f'Request Already Raised, please find the Request Details
here')
      else:

        serv_details = Services.objects.create(Serv_Type=serv_type, Serv_Name=serv_name,
Serv_State=state,
                                    Serv_City=city, Serv_Applied_By=requested_by,
                                    Serv_Location=location, Serv_Approval_Status='pending',
                                    Serv_Completion_Status='NA', Serv_Applied_Date=applied_date,
                                    Serv_Doc_Name=doc_name, Serv_Doc_Proof=doc_proof,
                                    Serv_Description=desc, Serv_Req_Image = Image)


        serv_details.save()
        messages.warning(request,'Service Request Submitted Successfully')
   return render(request, 'Services/Complaints.html')
```

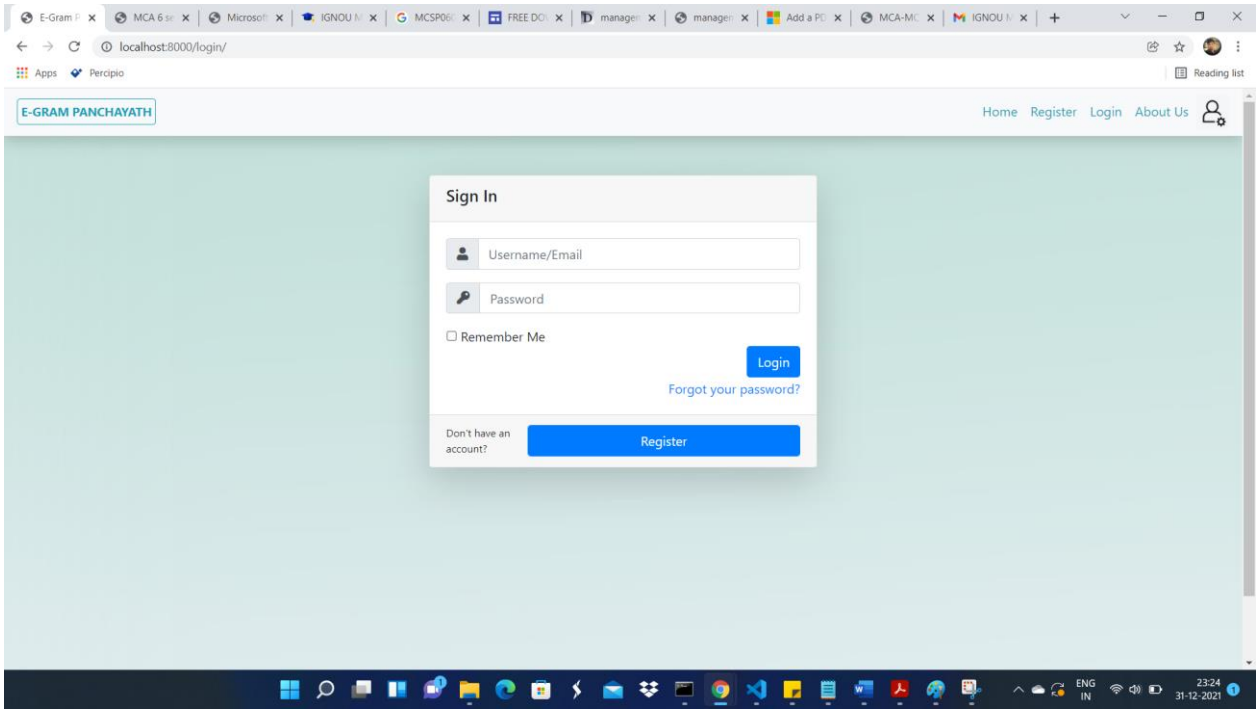
10. Graphical User Interface

➢ Home Page:

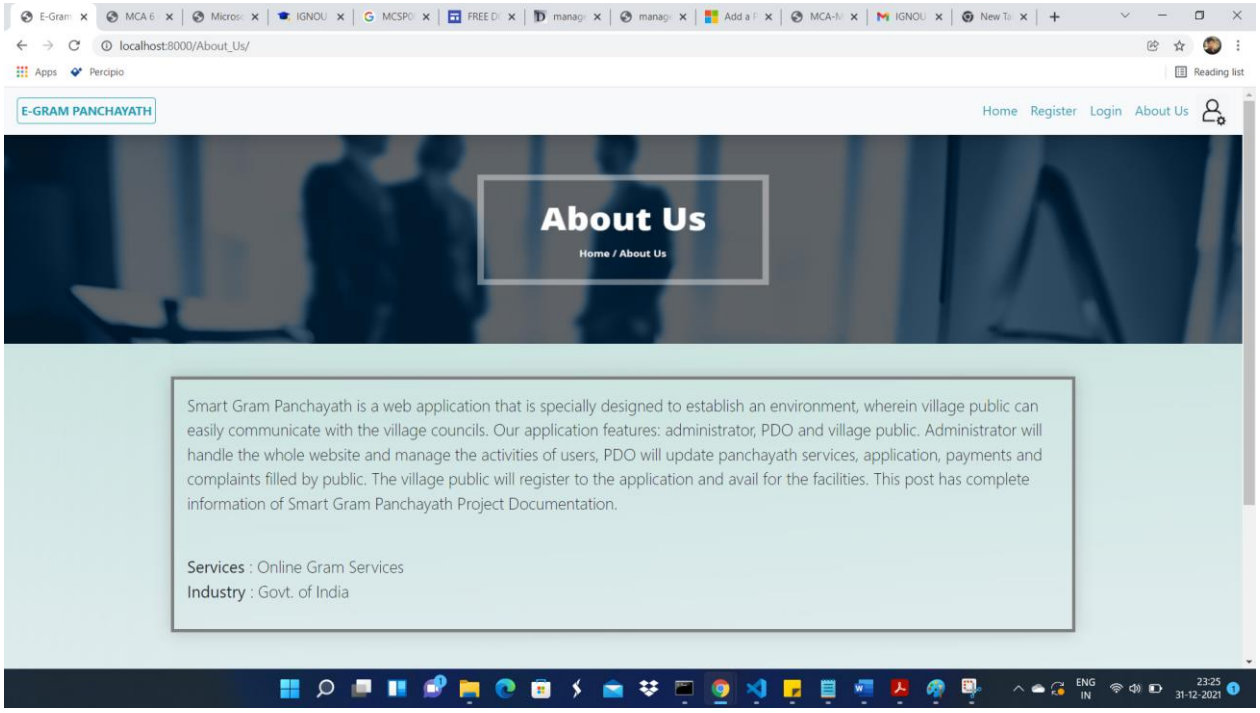
➢ Registration Page:



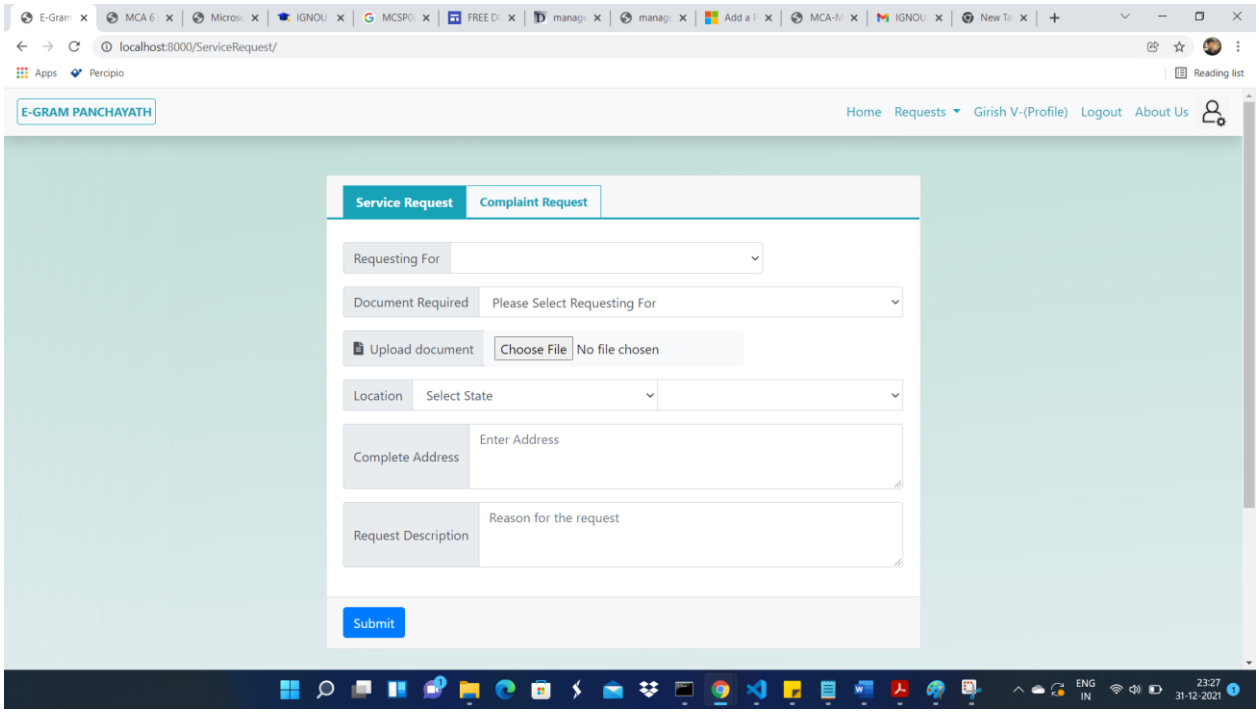
➢ Login Page:



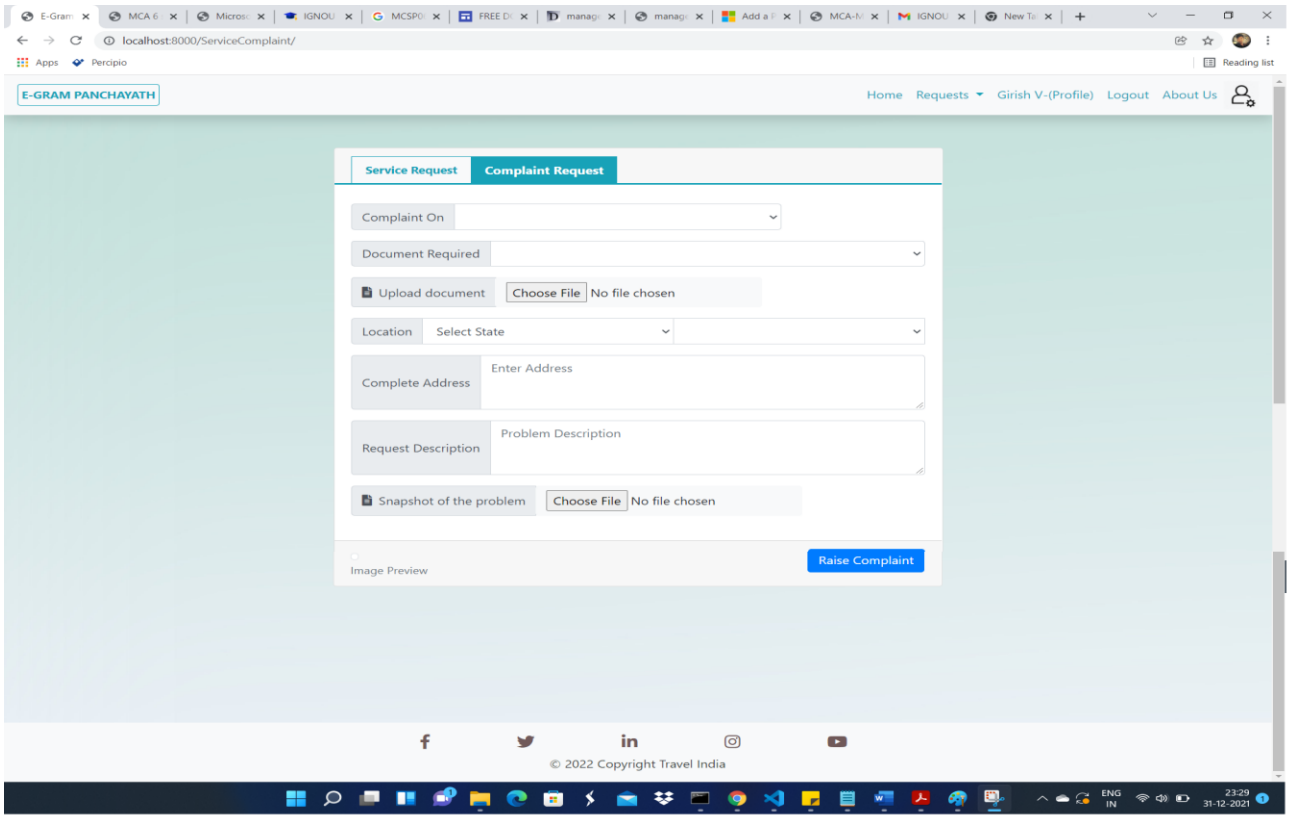
➢ About Us page:



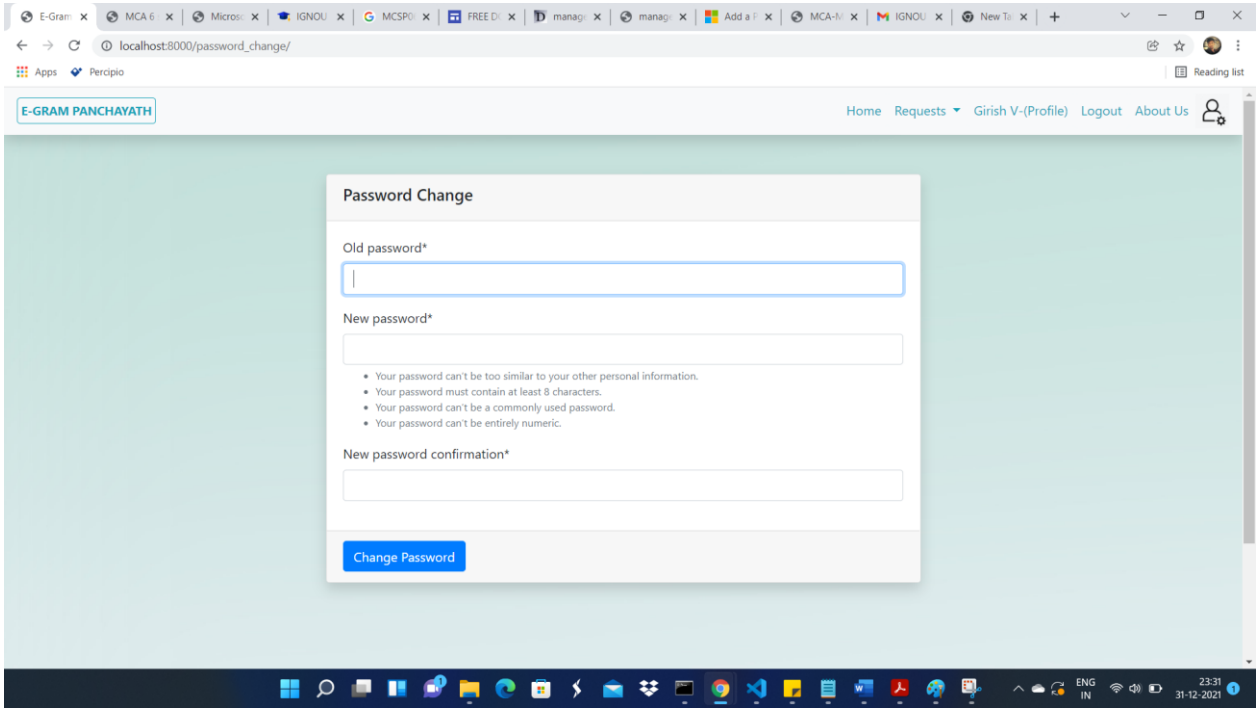
➢ Service Request Page:



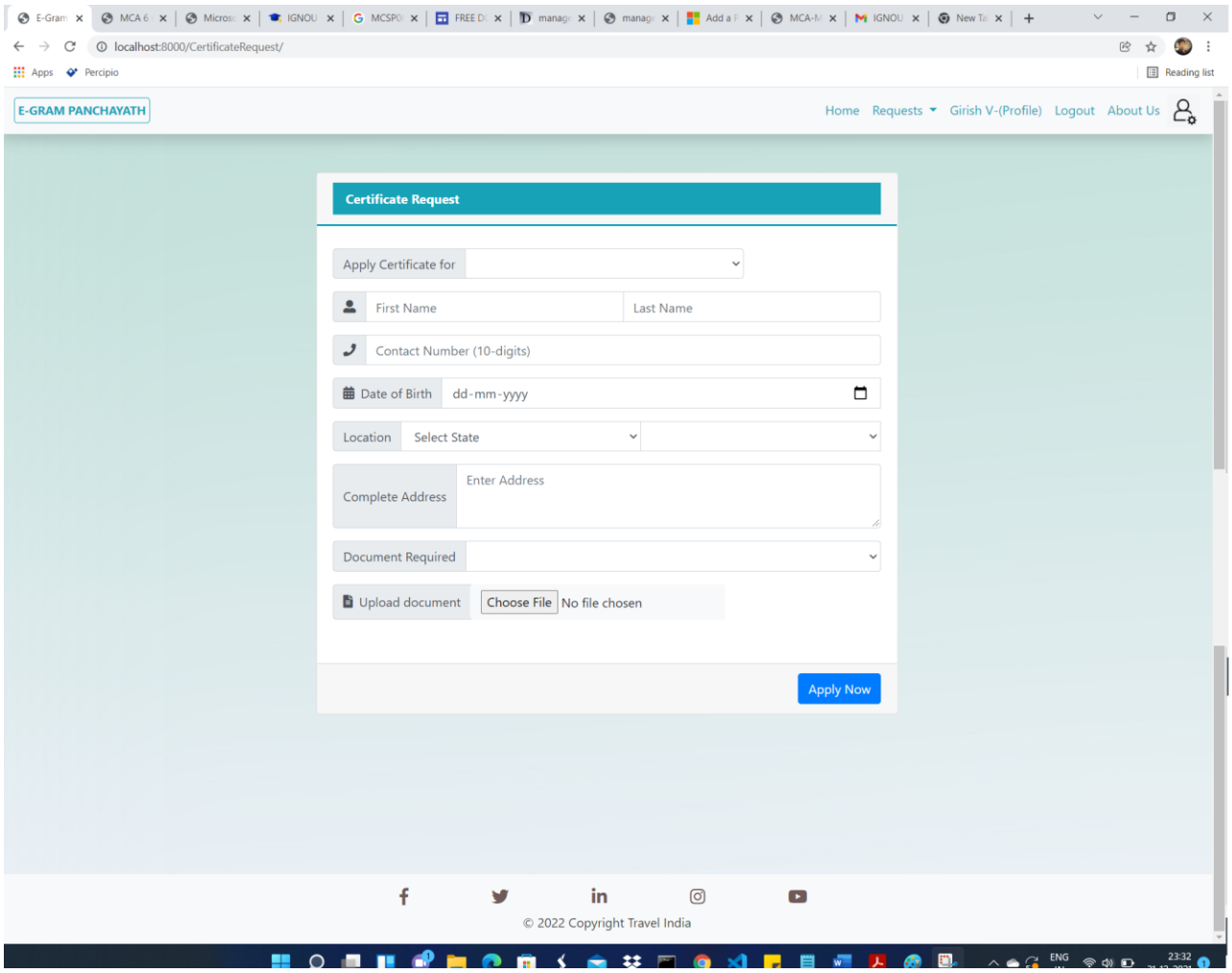
➢ Complaint Page:



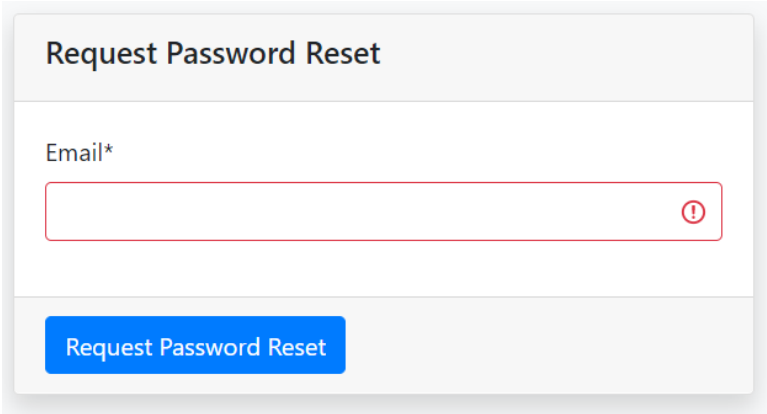
➢ Password Change Form:



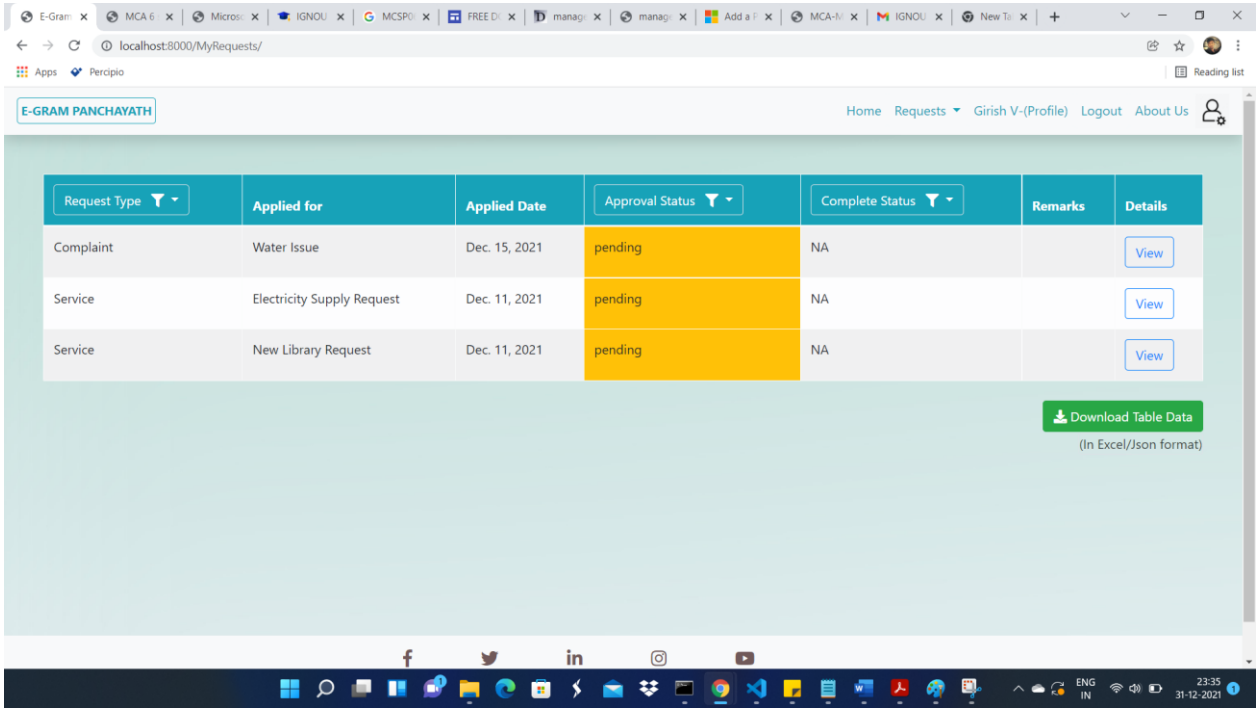
➢ Certificate Request page:



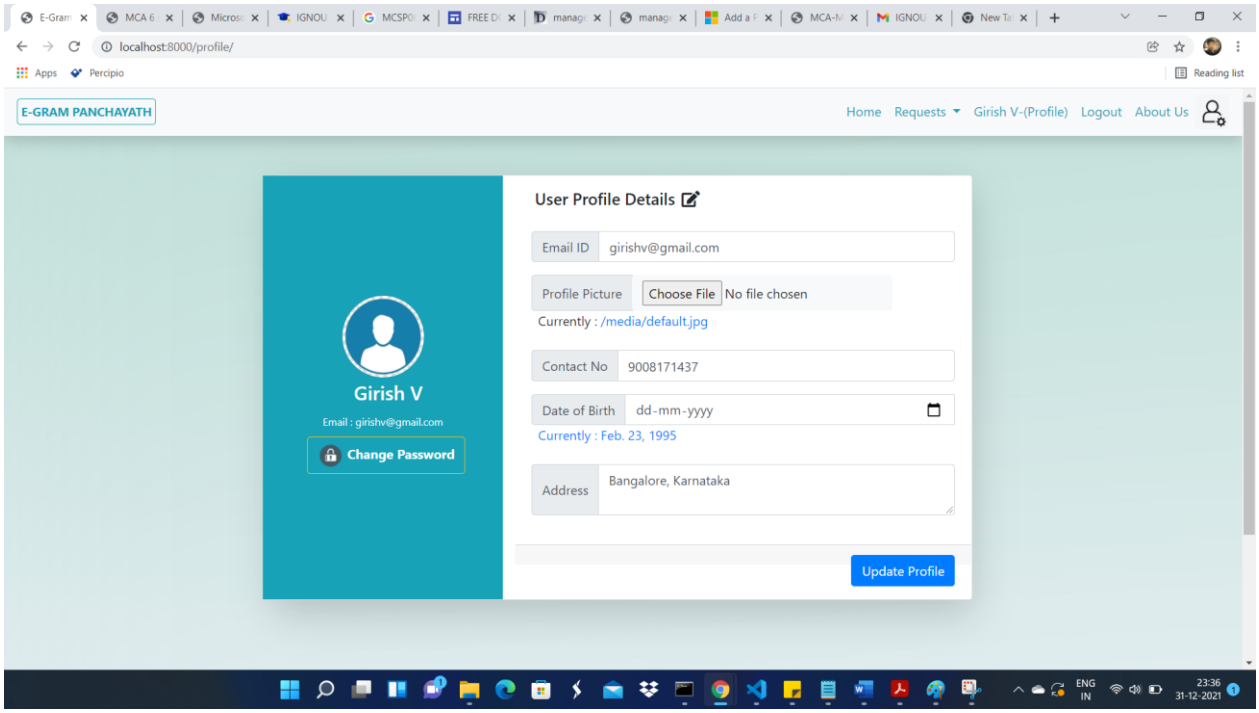
➢ Forgot Password Form:



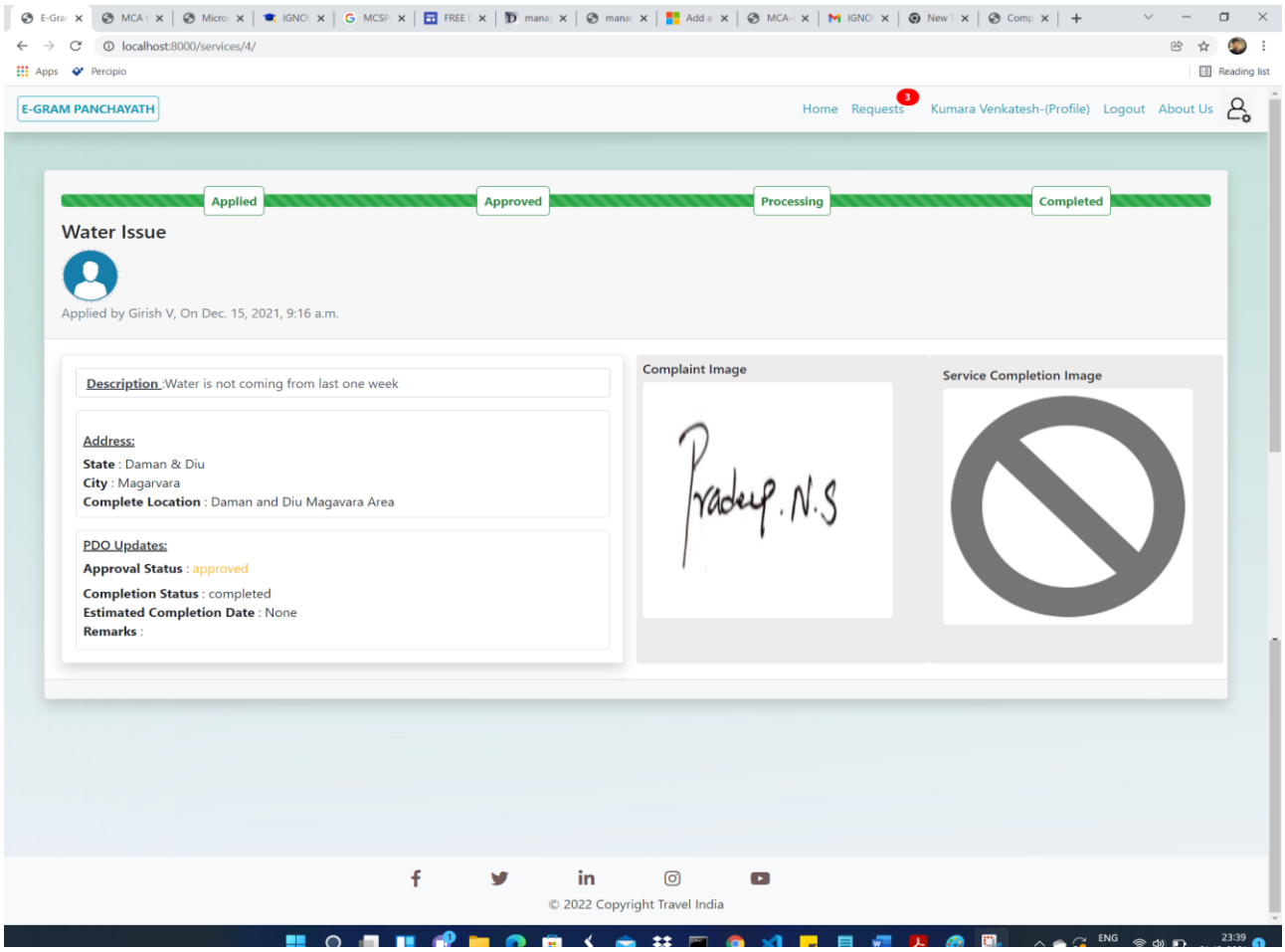
➢ My Requests Page:



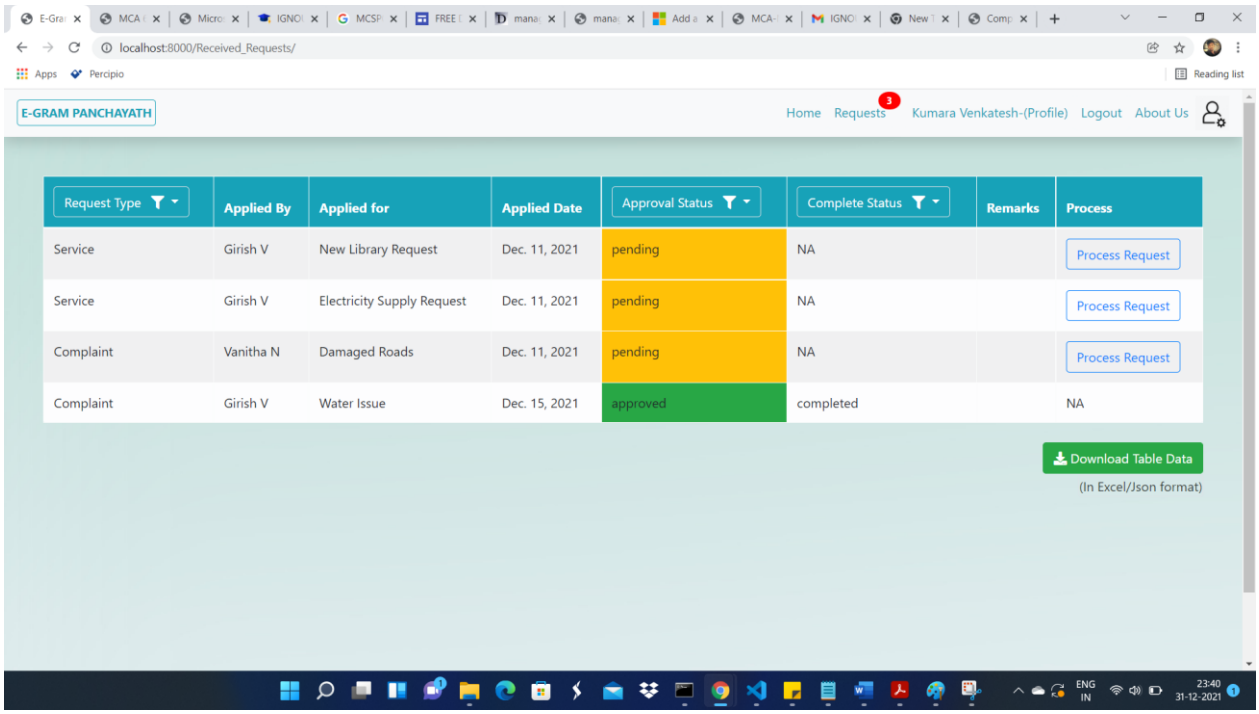
➢ Requests



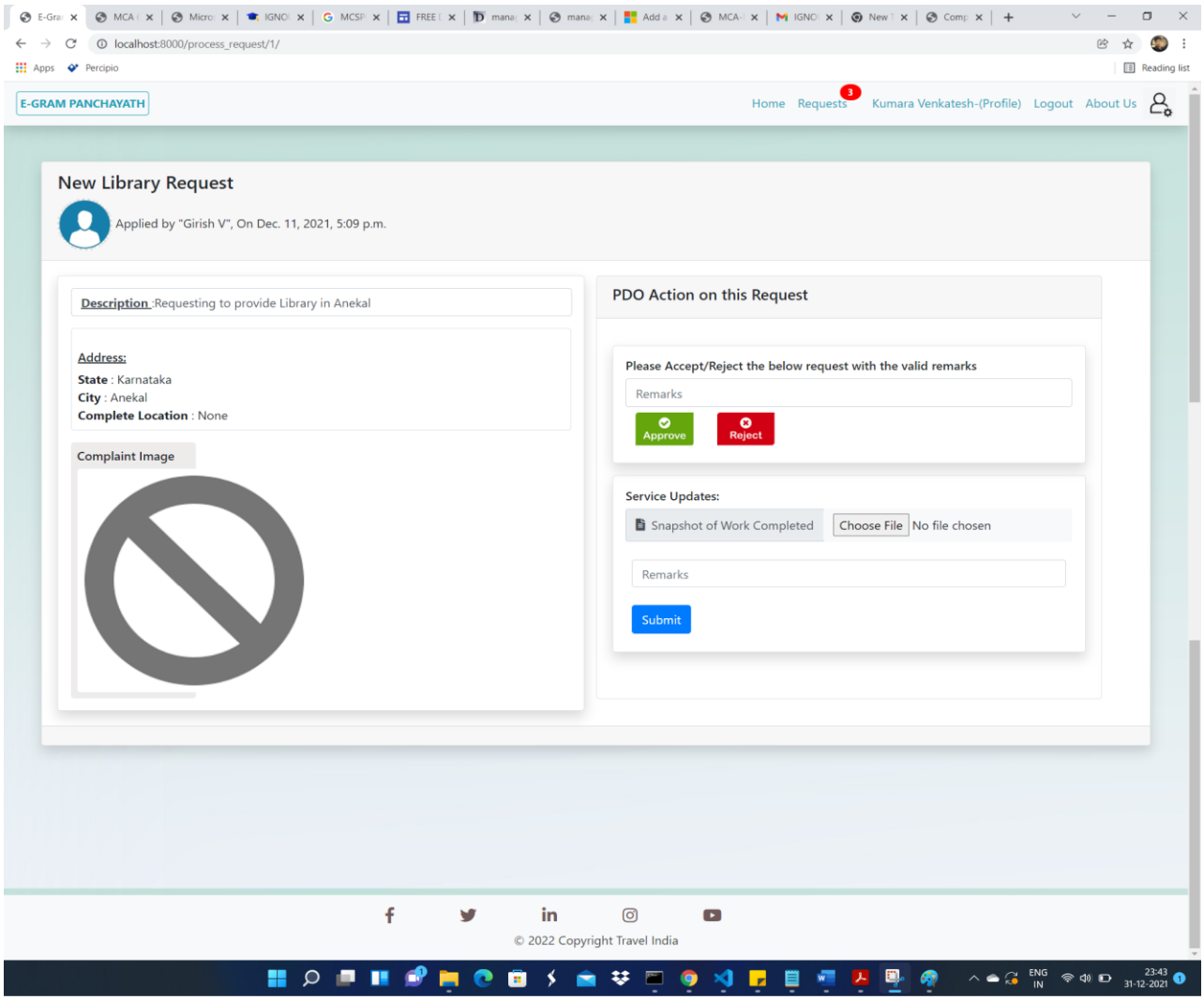
➢ Request Details:



➢ PDO Request Page:



➢ Process Requests Page:



11. TESTING

- **BLACK BOX TESTING**

In clearing house across various modules this testing was performed to check the following.

a) Establishing communication with the database for handling request and response.
b) Verification of OLE-DB providers (ADO) in functionality
c) Parameters passing and report generation used from the application with crystal report.

- **WHITE BOX TESTING**

All the statements included in the code across various modules were tested to find none of the statements were overlooked or skipped from execution. This enabled isolating of errors that would have otherwise occurred and would have resulted in abnormal terminal or exceptions thrown. The test was corely tested in patient and responsibility, insured party, ailments, procedures and applied payment modules.

- **STRING TESTING**

The applications were tested for inputs pertaining to patient data, responsible party, insured party for strings such as name, relation, employ information, policy details, insurance company details, claim center information and attorney data physician, reference physician information were tested for the following

a. null data

b. string length

c. data format

d. alpha numeric characters

In addition, numeric inputs were tested for invalid characters, invalid data format, size of the input data and the data type being handled.

- **UNIT TESTING**

Module pertaining to patient, responsible party, and soon were tested individually to check if the system performed the business logic or processors for the inputs provided and effective communication with the data base, the units were tested to check whether the data were reflected and updated across other tables that were used by other modules. The core modules

1. Responsible party and patient

2. Insured party

3. Ailments

4. Procedures

5. Applied payments

Were tested for the availability of data from other modules.

All the units were found to execute independently and had appropriate communication with the data base. Dependent modules were tested with static data and were found to execute as per SRS.

- **INTEGRATED TESTING**

All the units were combined from a menu driven application which then provided for integration with other modules the following well tested.

1. Message passing and communication between the modules

2. Data usage and synchronization

3. Flow of control using top-down testing confirming appropriate return of control as well as associated usability features.

- **SYSTEM TESTING**

The system as a whole along with required external resources was executed to check the dependencies, exception across the unavailability of the resources pertaining to the network connection, OLEDB providers, authentication of database and database itself.

DSN less connection and its effective communication for database was found to be as per their SRS.

- **MUTATION TESTING**

All fields across every module were tested rigorously with inputs that were intentionally provided with wrong data. This testing resolves bugs and errors through exception handling. That was a result of any kind of invalid data.

9. Future Scope & Further Enhancement

Further Enhancements: -

- This project has more scope in future and it can be integrated further
- This project is successfully implemented with all the features mentioned earlier.
- This project is designed keeping in view the needs of the common user and satisfying the user up to the maximum extent possible.
- This System being web-based and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.
- Therefore, we are successfully able to reach the goals and target of the project.

Conclusion: -

E-Governance for panchayat provides online services to the people living in that panchayat. It helps for the people in that area to easily complete their work which involves the action of authority of the panchayat people. As everything is made online people can request their applications from anywhere at any time. After requesting the certificate, the process will be carried out normally, no need for the people to go to panchayat office every time for the completion of work. It saves people time and they can perform their daily work without any interruption.

We have given the privileged modifications, changes and deletion. This information can be easily made available to all the stores throughout the offices through intranet/ Internet.

.

10.Bibliography

- Ref – Django: https://www.djangoproject.com/
- Ref – HTML: https://www.tutorialspoint.com/html/index.htm
- Ref – Bootstrap: https://www.w3schools.com/bootstrap/
- Ref - Python Study Materials: https://realpython.com/
- Ref- MySQL- https://www.mysqltutorial.org/
- Ref - Software Engineering –Roger S. Pressman

The End