A  
PROJECT REPORT  
ON

Freelancing Site

By

SHINGALA SHUBHAM P. (CE-143) (19CEUOS159)UNAGAR KEVAL V. (CE-164) (19CEUBG010)

VAGHANI SMIT D. (CE-166) (19CEUEG022)

B.Tech CE Semester - VSubject : Advanced Technologies (CE - 515)

Guided by:

Prof. Prashant M. Jadav

Prof. Siddharth P. Shah



**Faculty of Technology  
Department of Computer Engineering  
Dharmsinh Desai University**

  
**Faculty of Technology  
Department of Computer Engineering  
Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical / term work carried out in subject

Of **Advanced Technologies** ( CE - 515 ) and recorded in this

journal is the Bonafide work of

SHINGALA SHUBHAM P. (CE 143) (19CEUOS159)

**UNAGAR KEVAL V. (CE 164) (19CEUBG010)**

**VAGHANI SMIT D. (CE 166) (19CEUEG022)**

Of B.Tech semester **V** in the branch of **Computer Engineering**

During the academic year **2021-22.**

Prof. Prashant M. Jadav Dr. C. K. Bhensdadia,

Prof. Siddharth P. Shah Head,

Dept. of Computer Engg., Dept. of Computer Engg.,

Faculty of Technology Faculty of Technology

Dharmsinh Desai University, Nadiad Dharmsinh Desai University, Nadiad

**Contents:**

1 Introduction...……………………….………………..…….….……………………. 4

2 Software Requirement Specifications.…………..……………..…………….. 6

3 Design….…………………………….……………………..…….….………..………. 10

3.1 Data Flow Diagram………….……………………….……..…………… 12

3.2 E-R Diagram……………………………………………………………….… 13

4 Implementation Details ……………………………….…………..……….……. 18

5 Testing ……………………………......……..………………………….…………... 21

6 Screen-shots ……………………………………………………….………………… 26

7 Conclusion …………………………………………….………………...…….……. 27

8 Limitation and Future Extension ……………………………………………… 29

9 Bibliography ……………………………………………..…………………………… 27

1. **Introduction**

Freelancing Site is a online website between

* **Tools/Technologies Used**
* **Technologies:**
* Django
* Python
* MySQL
* Bootstrap
* CSS
* HTML
* **Tools**
* Git
* Visual Studio Code

* **Platform**
* Local development server

1. **Software Requirement Specifications**

**Users of the System :-**

* Citizens of India and NRIs ( Non-resident Indians )  
  BLO ( Booth level Officer )  
  ERO ( Electoral Registration Officer )  
  DEO ( District Election Officer )

**Functional requirements :-**

**1. Login / Registration**

**R.1.1: Create New Account  
Description :-** If user does not have an account on website then he/she has to create an  
Account.  
**Input:-** Enter details like name, id, password, etc... to create an account.  
**Output:-** Display success message.

**R.1.2: Login  
Description :-** User can access this website by login using id and password.  
**Input:-** Enter id and password.  
**Output:-** If id and password are correct then all functionalities will be display, otherwise message  
will be display as “Invalid username or password”.

**To Use Any Of the below module & functionalities user must require login.**

**2. New Application / Modification**

***R.2.1: Apply for new voter id card*Description :-** User can apply for new voter id card ( Age >= 18 ) by filling form and uploading  
required documents. if all the details and documents are verified then user will get voter  
id card at provided address within 30 days.

**R.2.1.1: Form Details  
Input:-** User have to enter all required details.  
**Output:-** Display Success message and user will be redirected to next page.  
**Next function:-** Upload documents.

**R.2.1.2: Upload Documents  
Input:-** User has to upload all the required documents.  
**Output:-** Display Success message and user will be redirected to next page.  
**Next function:-** Preview details

**R.2.1.3: Preview Details & Submit  
Input:-** User can see all the details and uploaded documents, if everything is correct then submit  
otherwise edit details.  
**Output:-** if submitted then display success message otherwise redirected to form details in edit  
mode.

***R.2.2: Modification in voter id card***  
**Description :-** User can apply to modify personal details like name, address, phone no, etc…  
**Input:-** Enter EPIC ( Electoral Photo Identity Card ) number of voter id card.  
**Output:-** If EPIC number matches in database than user can apply to modify details, otherwise display  
“Invalid epic number” message.

***R.2.3: Document verification*Description :-** Officer Check the documents uploaded by user and verified it and update application  
Status.  
**Input :-** All request of new & modification application.  
**Output :-** if all documents are correct then application status updated as “document verified” else  
Application status updated as “upload valid documents”.

**3. Forms**

***R.3.1: Download Forms*Description:-** User can download form according to their requirements and submit it offline at office  
and do procced offline also.  
**Input:-** User have to select form according to their requirements.  
**Output:-** PDF file.

***R.3.2: Add form*Description :-** Officer(Admin) adds new form.  
**Input :-** Add form as pdf.  
**Output :-** Form added successfully.

***R.3.3: Remove/delete form*Description :-** Officer(admin) can delete the form.  
**Input :-** Select the form & delete.  
**Output :-** Form deleted successfully.

**4. Electoral roll**

***R.4.1: Search in electoral roll*Description :-** User can search their names in electoral roll by epic number or by name & address.  
**Input :-** Enter details.  
**Output :-** If details found then user can see their details in electoral roll else message display as “Not found”.

***R.4.2: Download electoral roll*Description :-** User can download electoral roll pdf by selecting state -> district -> assembly.  
**Input :-** Select state -> district -> assembly  
**Output :-** Pdf downloaded.

***R.4.3: Add/remove details in electoral roll*Description :-** Officer(admin) can add/remove details in any electoral roll pdf.  
**Input :-** Select and edit details.  
**Output :-** Details edited successfully.

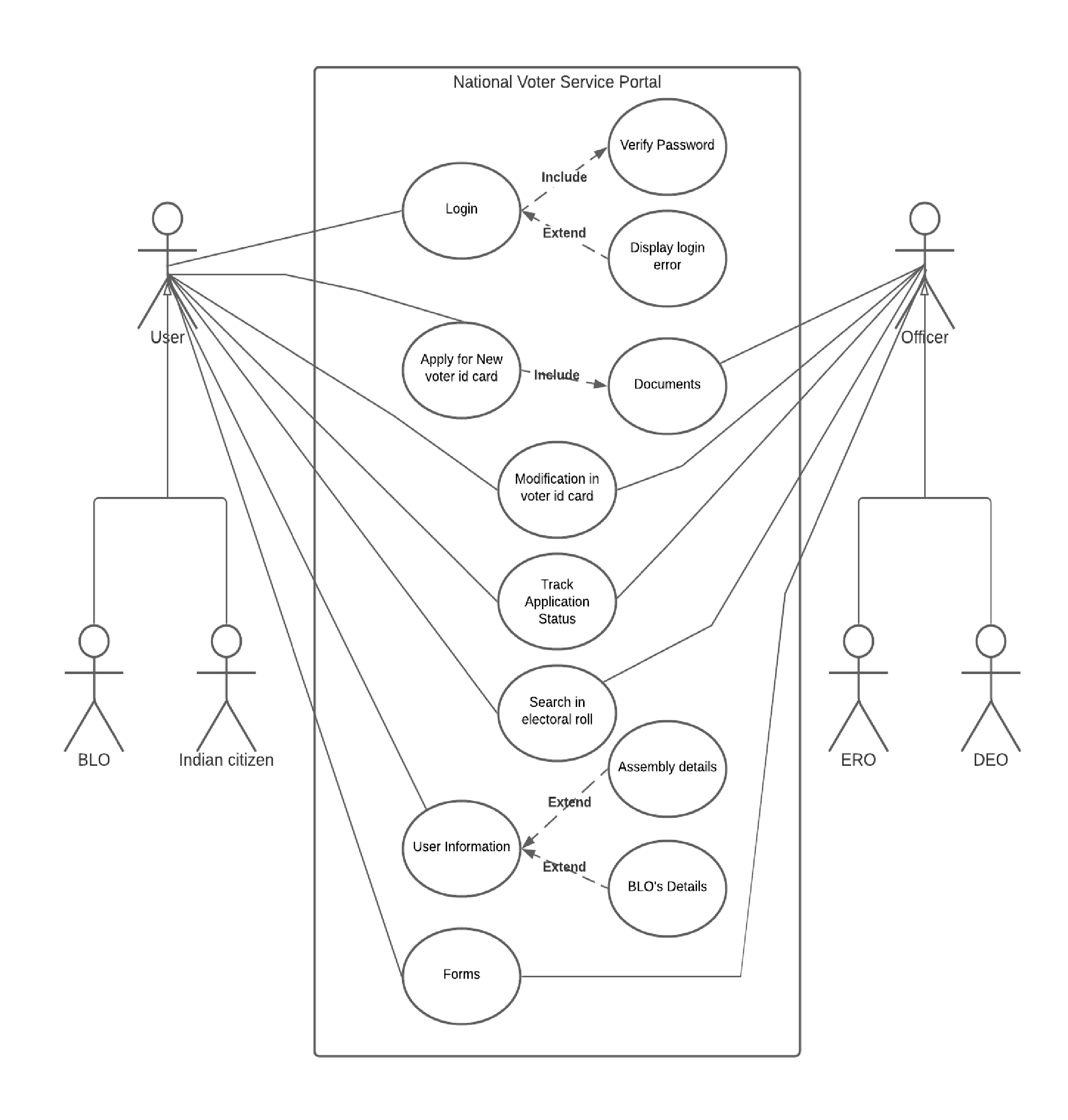
**5. User details & status.**

***R.5.1: Track application status*Description :-** User can track status of his/her voter id card application.  
**Input:-** Enter EPIC number of voter id card.  
**Output:-** If EPIC number matches in database then user can see their application status details  
otherwise display “Invalid epic number” message.

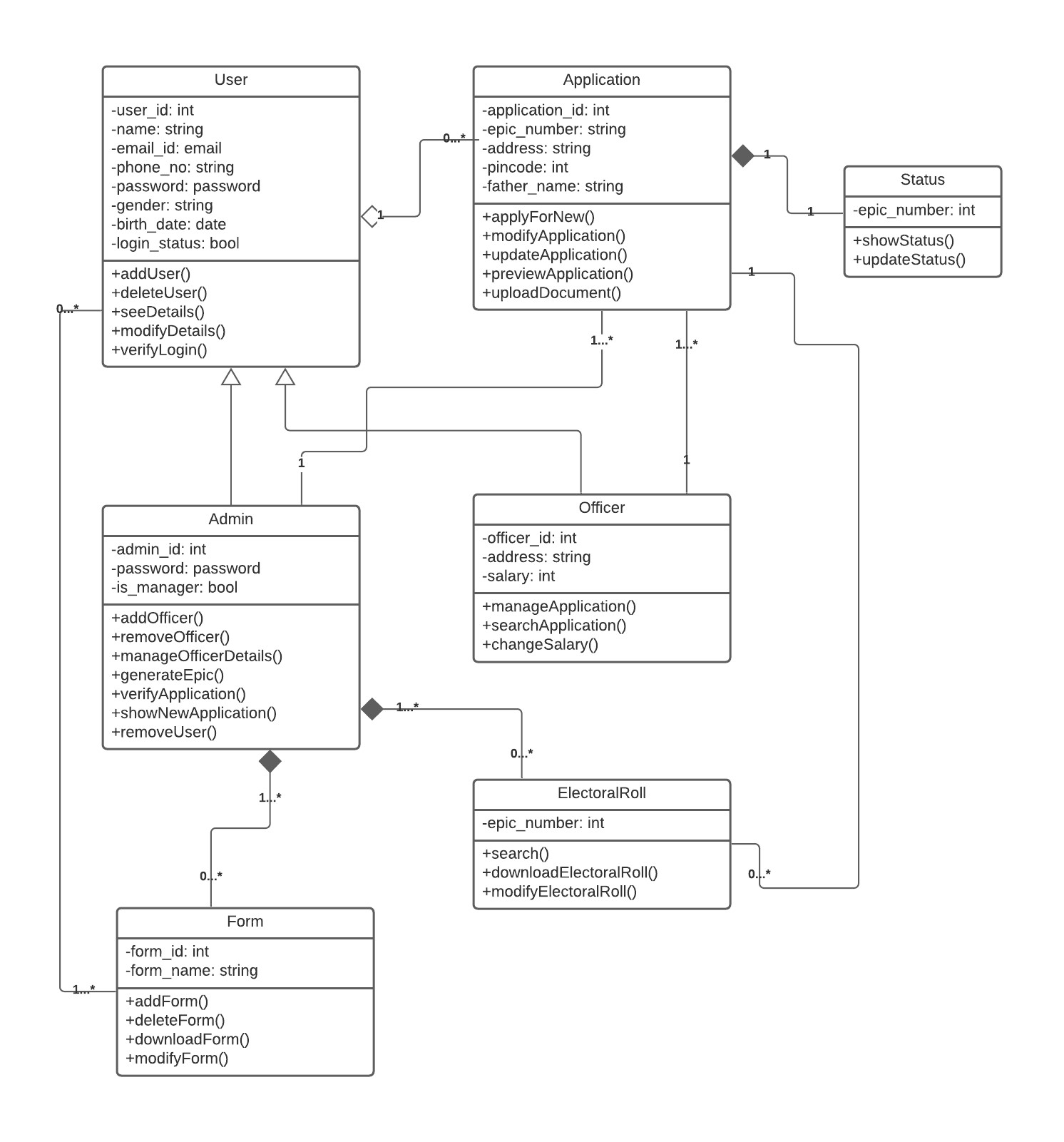
**R.5.2: Add required details & documents.  
Description :-** if user find some remark to add/update some thing on their application status then they  
have to Add that details or documents.  
**Input :-** Enter details or documents whichever is asked in application status.  
**Output :-** Successful message.

***R.5.3: Know Details*Description:-** User can see his/her details, assembly / parliamentary constituency details, BLO  
Electoral officer’s details, political party representative etc…  
**Input:-** Enter EPIC number of voter id card.  
**Output:-** If epic number matches in database then user can see all the details, otherwise display  
“Invalid epic number” message.

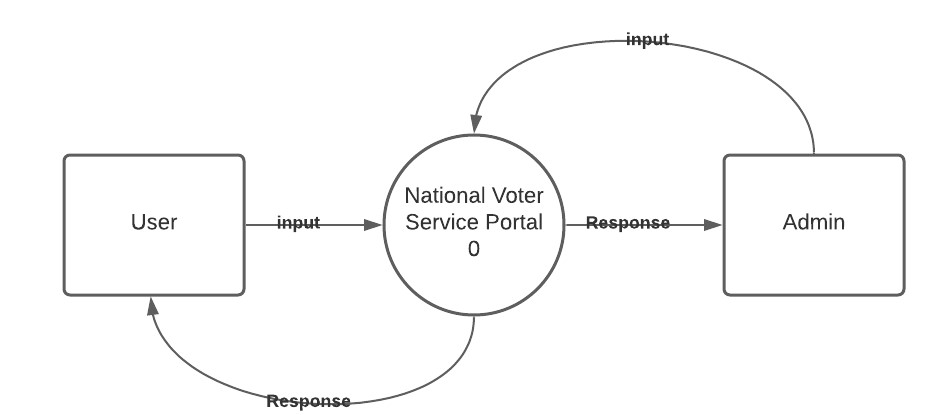
1. **Design** 
   1. **Use Case Diagram**



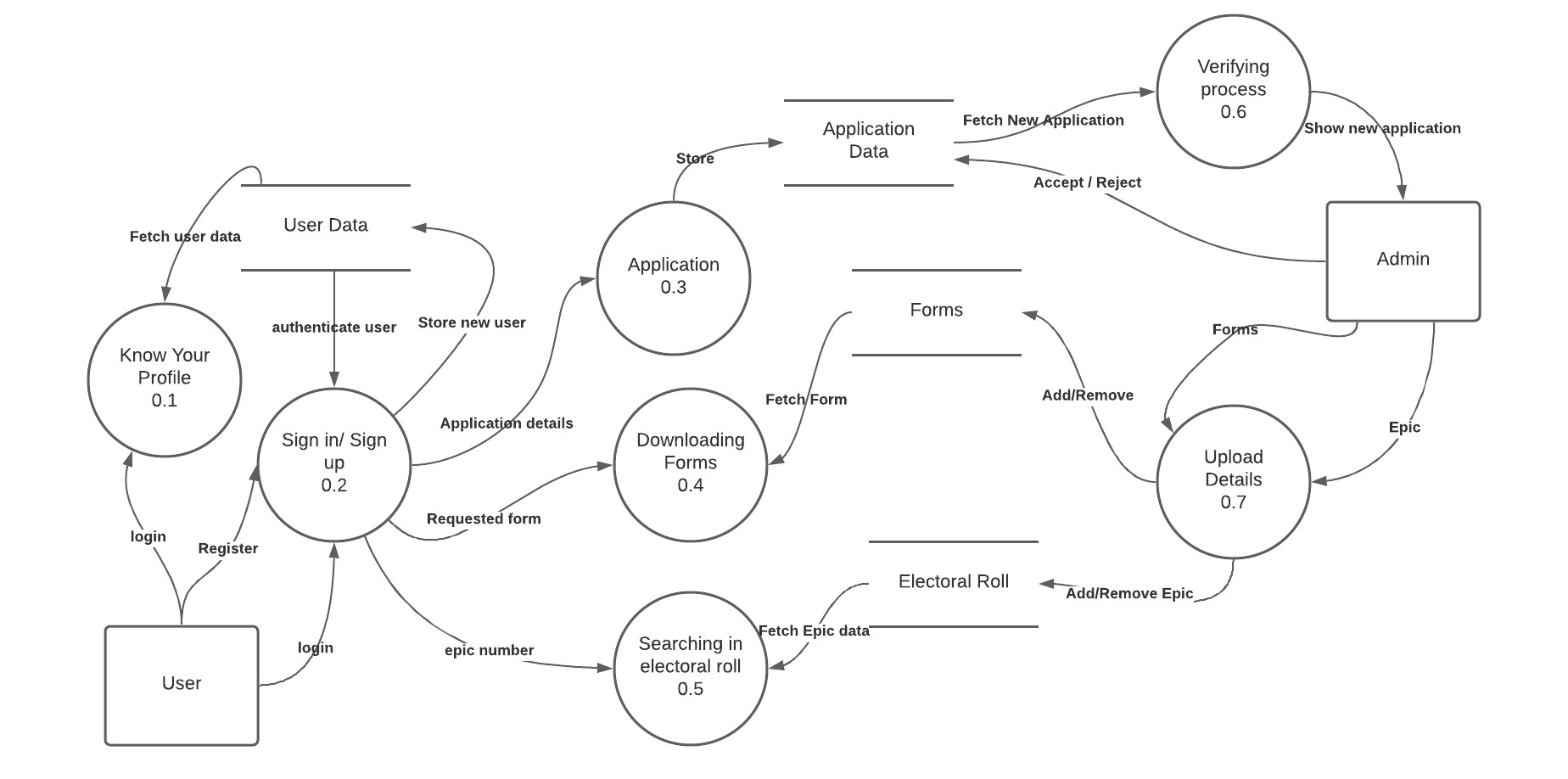
* 1. **Class Diagram**



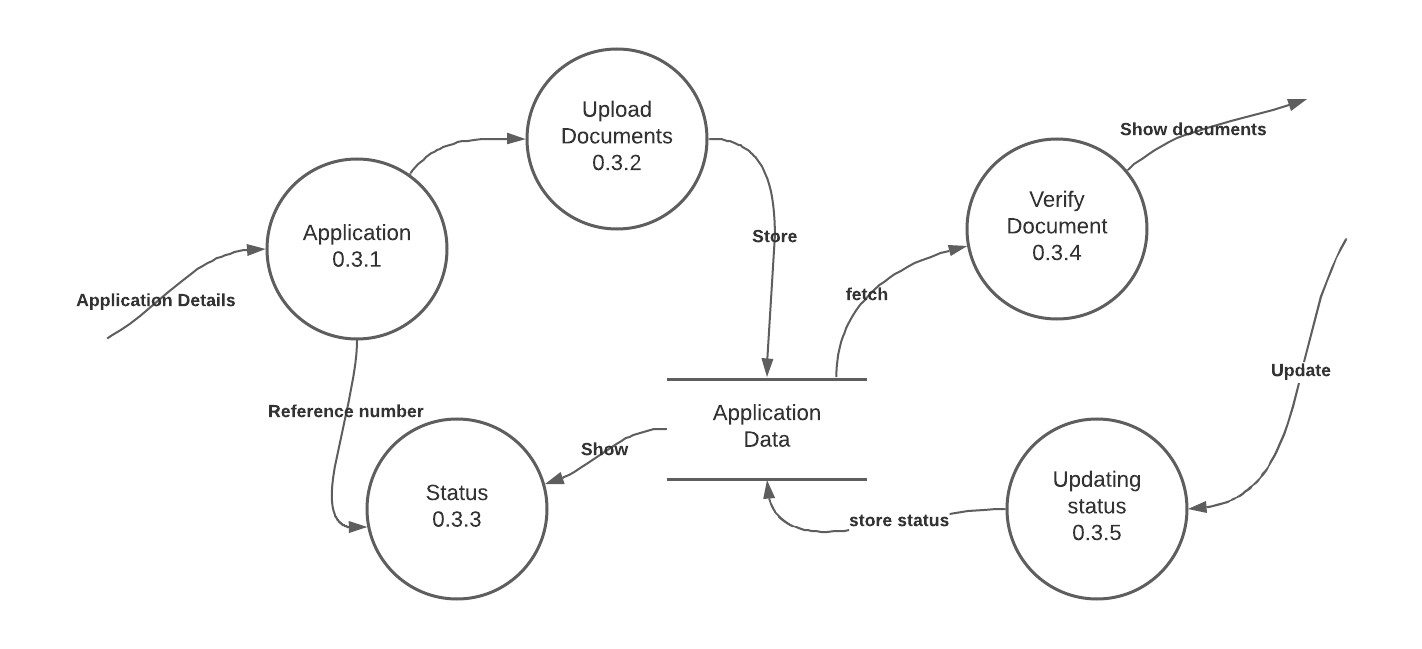
* 1. **Data Flow Diagram**



**Context Level Diagram**

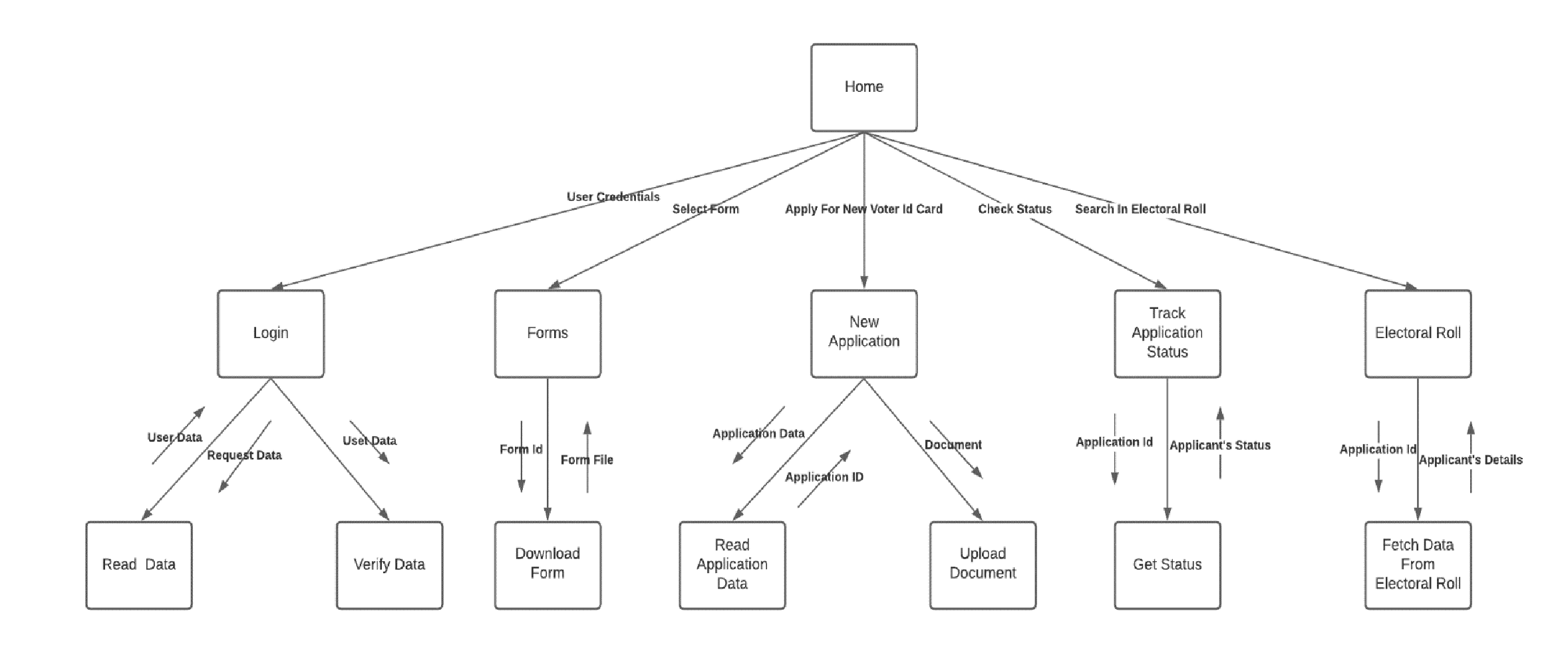


**Level - 1 Diagram**



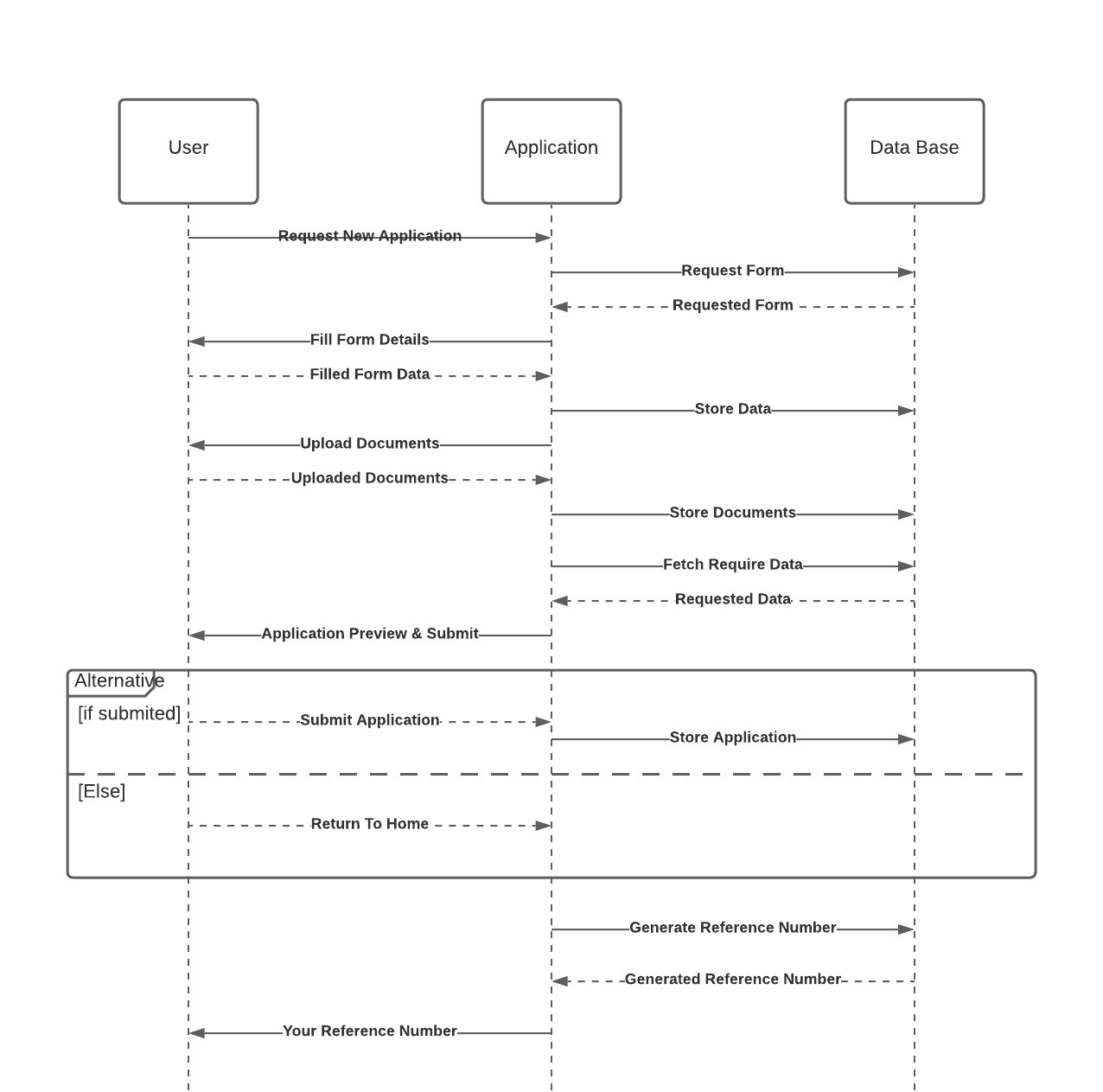
**Level - 2 Diagram**

* 1. **Structure Chart**

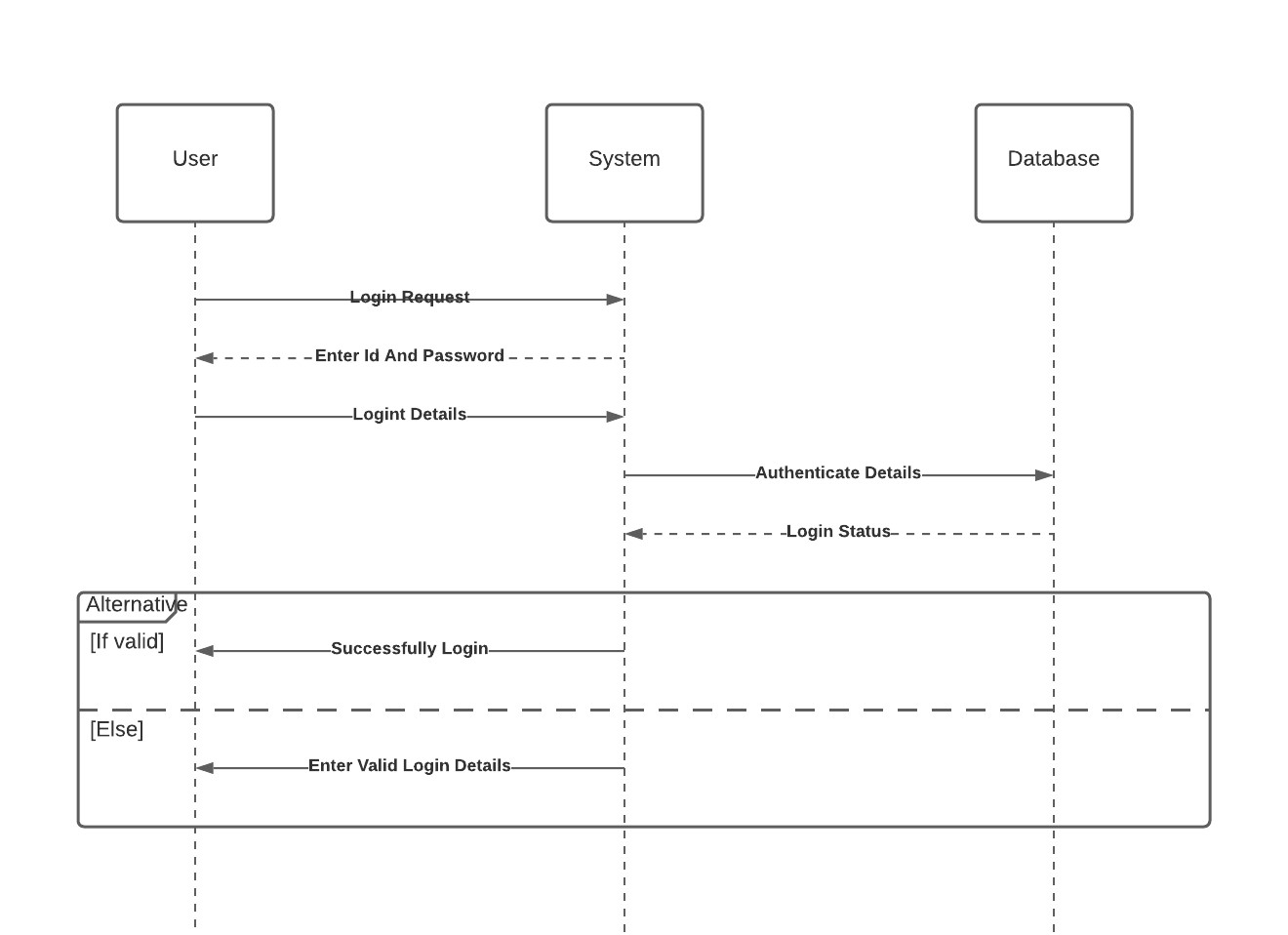


* 1. **Sequence Diagram**

Apply for new voter id card

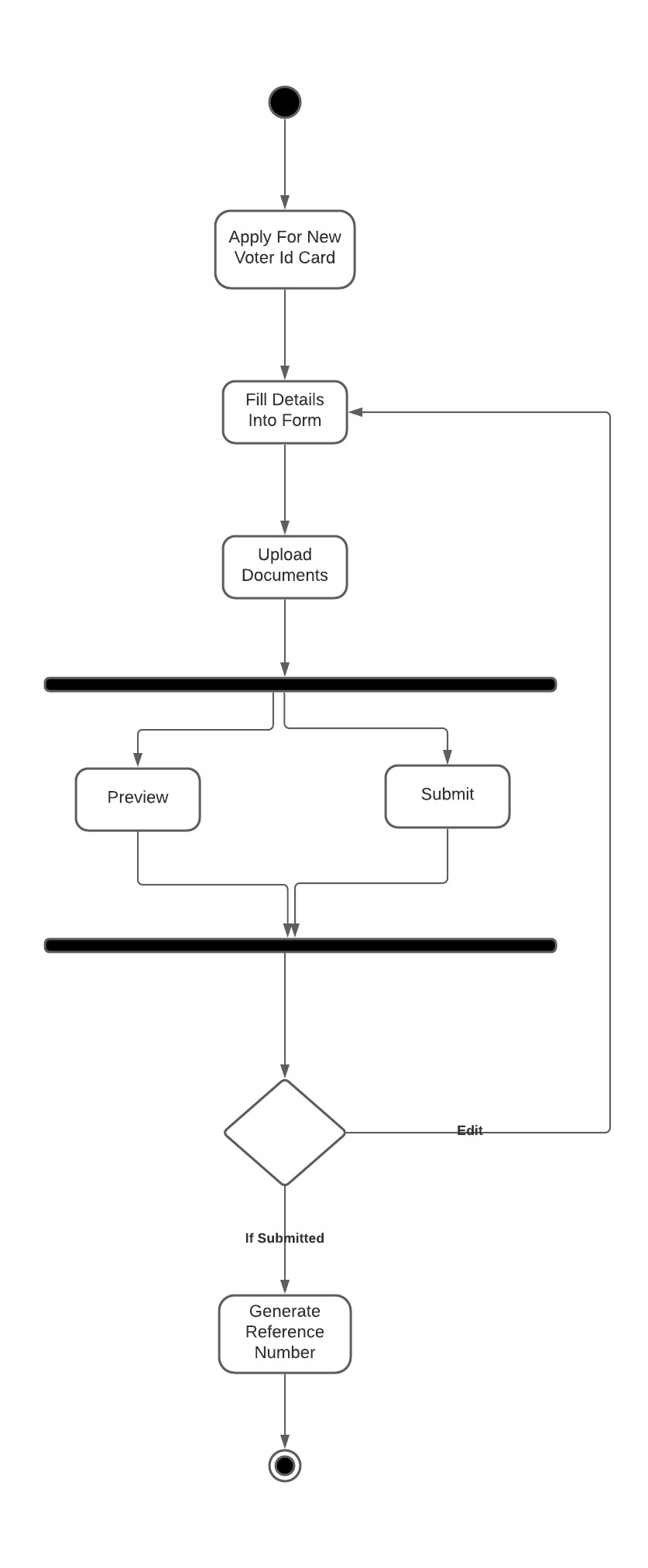


Login

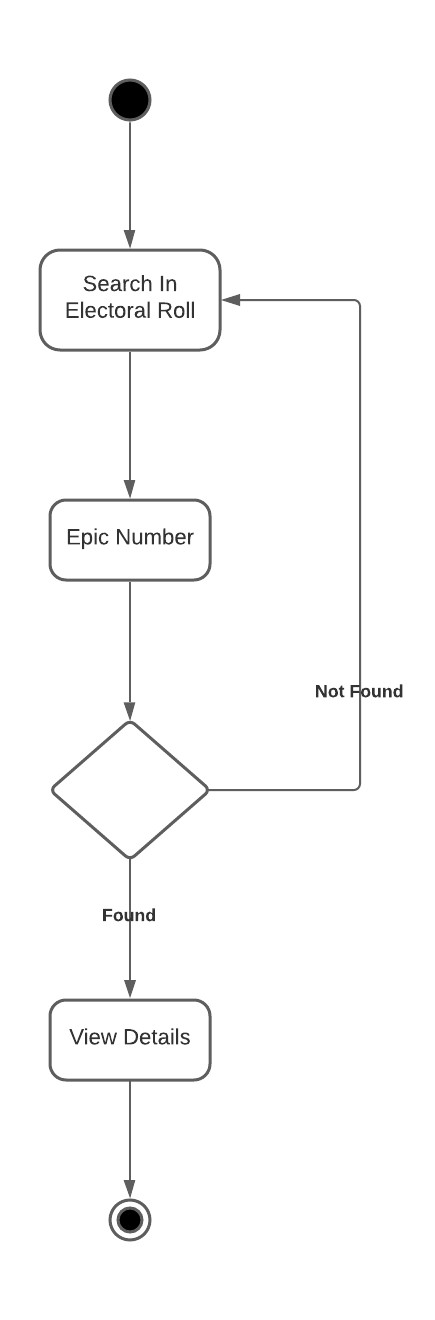


* 1. **Activity Diagram**

Apply for new voter id card



Login



1. **Implementation Details**

* Apply For New Voter Id Card :-

def application(request):

    today = date.today()

    if request.method=='POST':

        apply.name = request.POST['name']

        apply.surname = request.POST['surname']

        apply.gender = request.POST['gender']

        apply.state = request.POST['state']

        apply.city = request.POST['city']

        apply.address = request.POST['address']

        apply.pincode = request.POST['pincode']

        apply.dob = request.POST['dob']

        apply.relation = request.POST['relation']

        apply.relativeName = request.POST['relativeName']

        apply.relativeSurname = request.POST['relativeSurname']

        apply.email = request.POST['email']

        apply.mobile = request.POST['phone']

        apply.declarationPlace = request.POST['applyFrom']

        apply.declarationDate = request.POST['applyAt']

        apply.dob = datetime.datetime.strptime(apply.dob, '%Y-%m-%d')

        if  (today.year - apply.dob.year - ((today.month, today.day) < (apply.dob.month, apply.dob.day))) < 18 :

            messages.info(request,'Youe age is less than 18 you can not apply for voter id card')

            return redirect('application')

        else:

            #apply = ApplicationData(name=name , surname=surname , gender=gender , state=state , city=city , address=address , pincode=pincode , dob=dob , relation=relation , relativeName=relativeName , relativeSurname=relativeSurname , email=email , mobile=mobile , declarationDate=declarationDate , declarationPlace=declarationPlace)

            apply.save()

            return redirect('upload')

    else:

        if request.user.is\_authenticated:

            return render(request,'applicationform.html')

        messages.info(request,'Please Login first then apply for new application')

        return redirect('/login')

* Upload Documents :-

def upload(request):

    if request.method == 'POST':

        apply.photo = request.FILES['photo']

        apply.aadharCard = request.FILES['aadharCard']

        apply.declaration = request.FILES['declaration']

        #apply = ApplicationData(photo=photo, aadharCard=aadharCard, declaration=declaration)

        apply.save()

        return redirect('preview')

    else:

        return render(request, 'upload\_documents.html')

* Preview Application :-

def preview(request):

    date=str(apply.dob.strftime("%d%m%Y"))

    apply.application\_id=(str(apply.name))+date

    apply.save()

    return render(request, 'preview.html', {'apply': apply })

* Track Application Status :-

def status(request):

    if request.user.is\_authenticated:

        if request.method == 'POST':

            id = request.POST['id']

            obj = ApplicationData.objects.filter(application\_id=id)

            return render(request, 'showstatus.html', {'obj': obj })

        else:

            send = "status"

            return render(request, 'applicationid.html', {'obj': send})

    else:

        messages.info(request,'Please Login first')

        return redirect('/login')

* Modification in Voter id card :-

if request.user.is\_authenticated:

        if request.method == "POST" and (request.POST['submit'] == 'Submit'):

            id = request.POST['id']

            obj = ApplicationData.objects.filter(application\_id=id)

            return render(request, 'modificationform.html', {'obj': obj })

        elif request.method == "POST" and (request.POST['submit'] == 'Modify'):

            id = request.POST['id']

            ApplicationData.objects.filter(application\_id=id).update(name = request.POST['name'])

            ApplicationData.objects.filter(application\_id=id).update(surname = request.POST['surname'])

            ApplicationData.objects.filter(application\_id=id).update(gender = request.POST['gender'])

            ApplicationData.objects.filter(application\_id=id).update(state = request.POST['state'])

            ApplicationData.objects.filter(application\_id=id).update(city = request.POST['city'])

            ApplicationData.objects.filter(application\_id=id).update(address = request.POST['address'])

            ApplicationData.objects.filter(application\_id=id).update(pincode = request.POST['pincode'])

            ApplicationData.objects.filter(application\_id=id).update(relation = request.POST['relation'])

            ApplicationData.objects.filter(application\_id=id).update(relativeName = request.POST['relativeName'])

            ApplicationData.objects.filter(application\_id=id).update(relativeSurname = request.POST['relativeSurname'])

            ApplicationData.objects.filter(application\_id=id).update(email = request.POST['email'])

            ApplicationData.objects.filter(application\_id=id).update(mobile = request.POST['phone'])

            ApplicationData.objects.filter(application\_id=id).update(declarationPlace = request.POST['applyFrom'])

            ApplicationData.objects.filter(application\_id=id).update(dob = datetime.datetime.strptime(request.POST['dob'], '%Y-%m-%d'))

            return redirect('/')

        else:

            send = "modification"

            return render(request,'applicationid.html', {'obj': send})

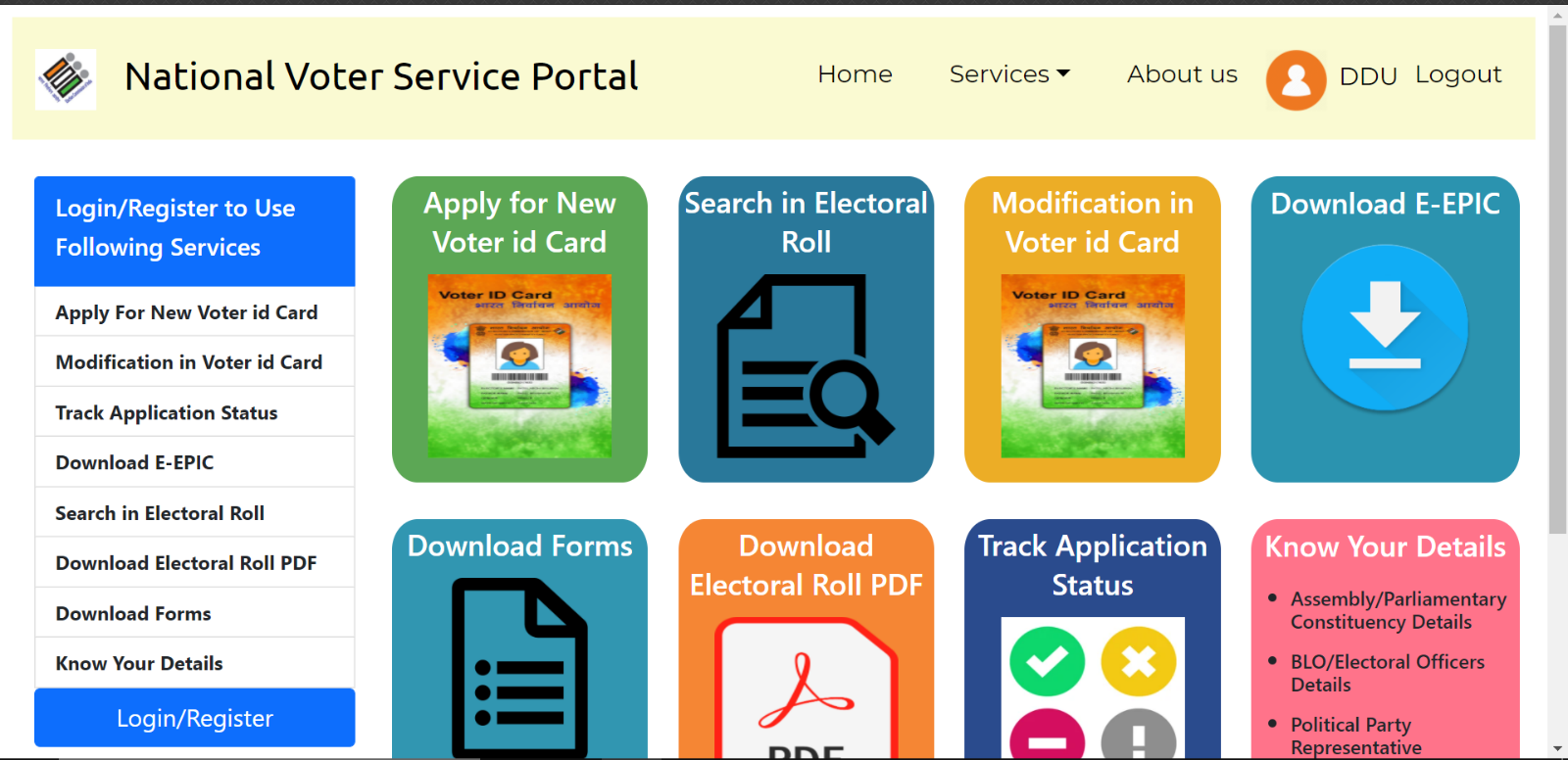
    else:

        messages.info(request,'Please Login first then apply for Modification')

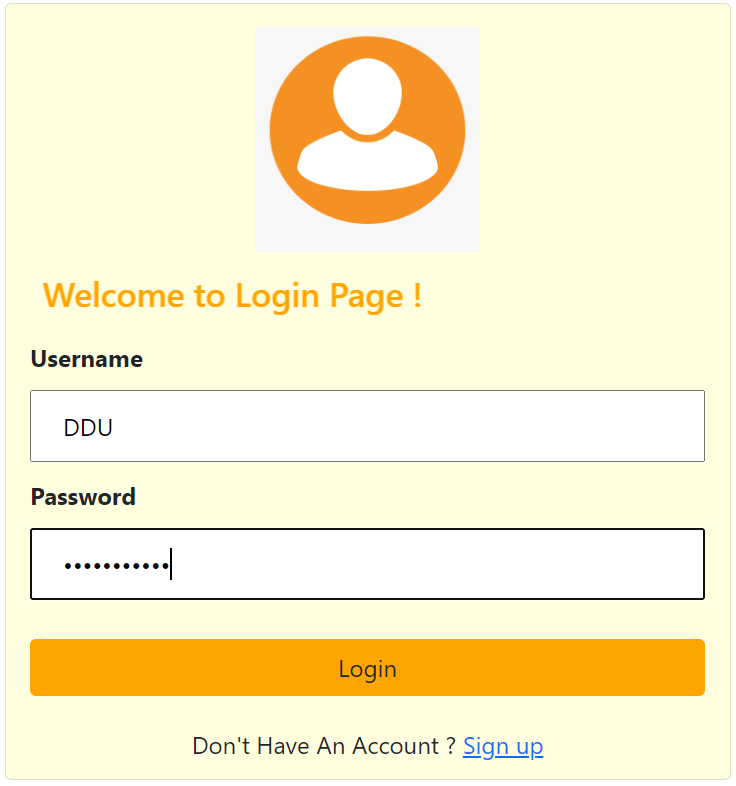
        return redirect('/login')

1. **Screenshots**

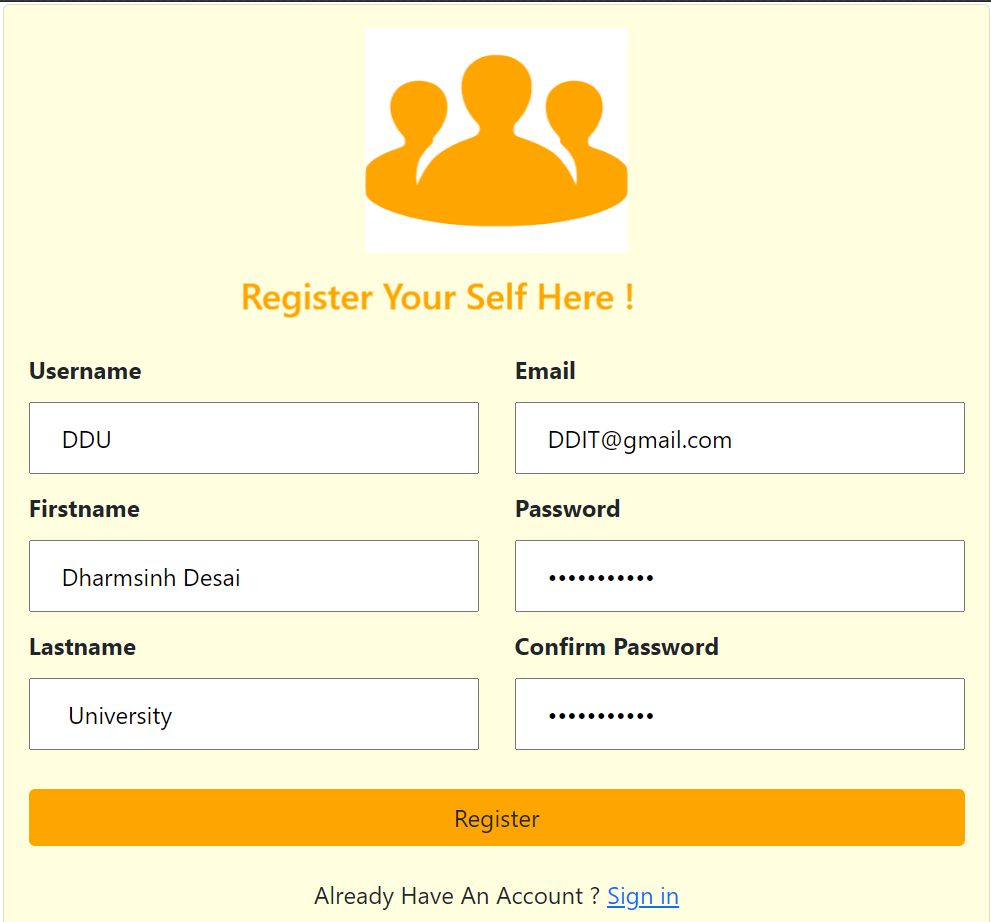
* Home :-



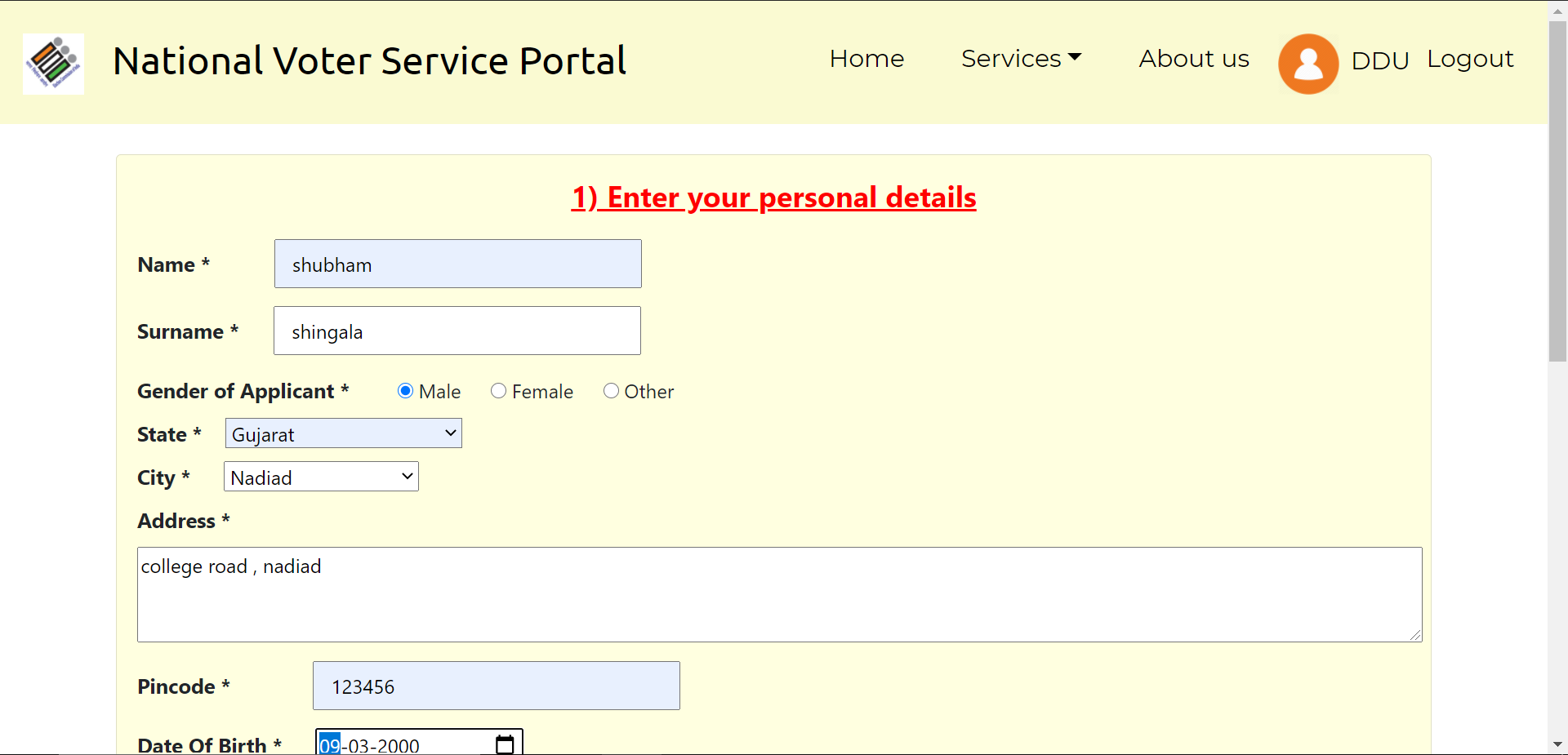
* Login :-

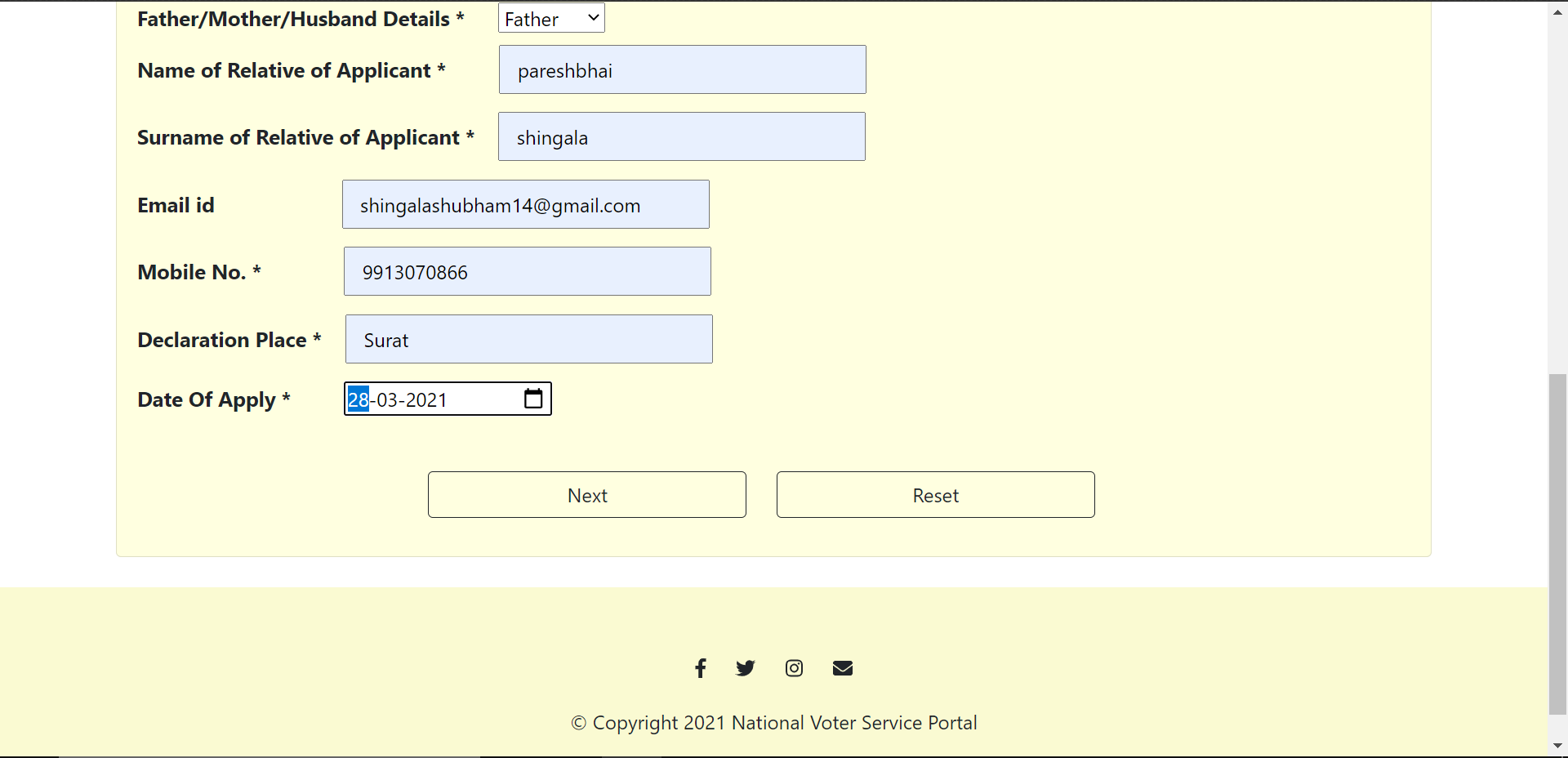


* Register :-

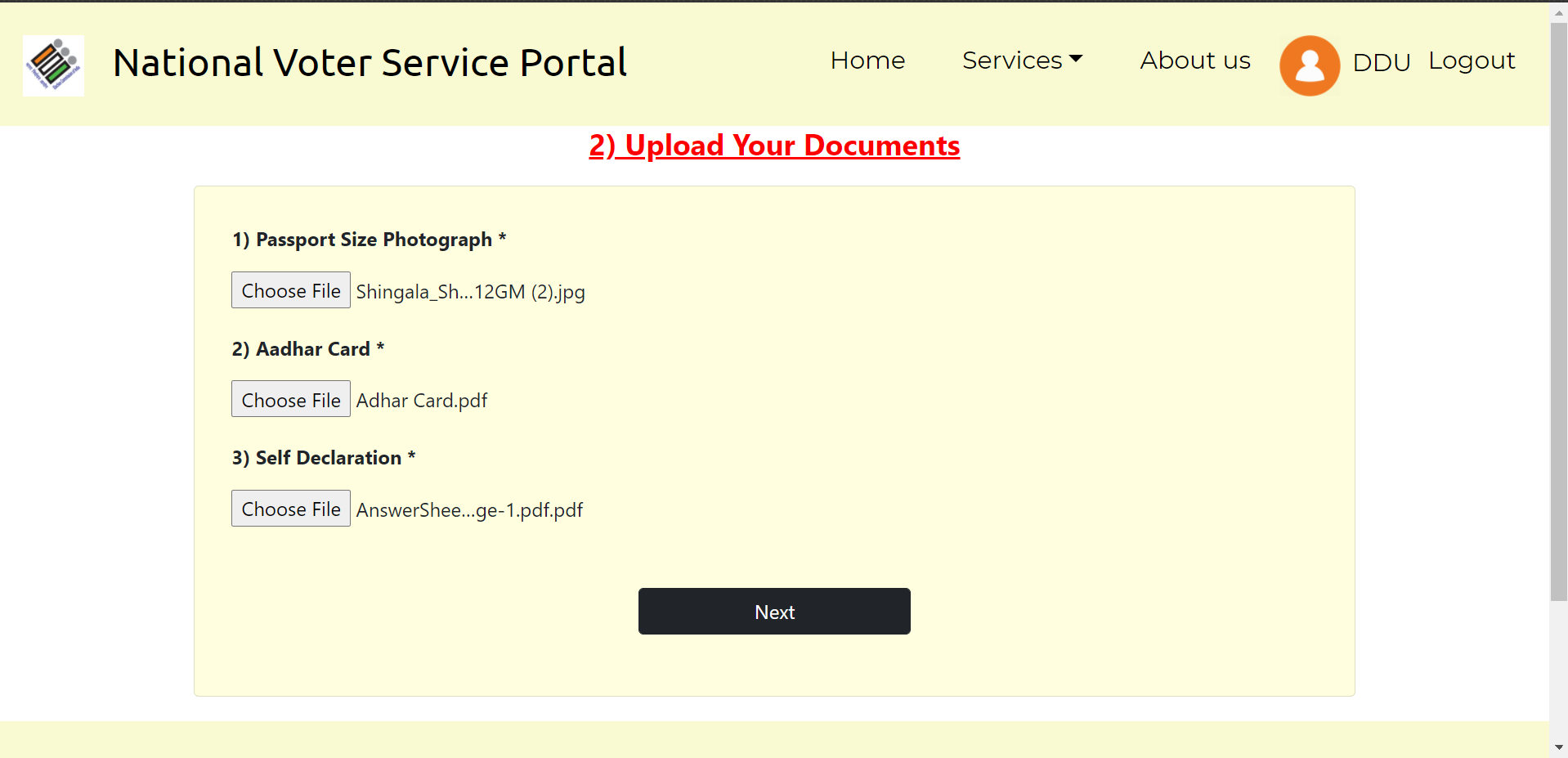


* New Application Form :-

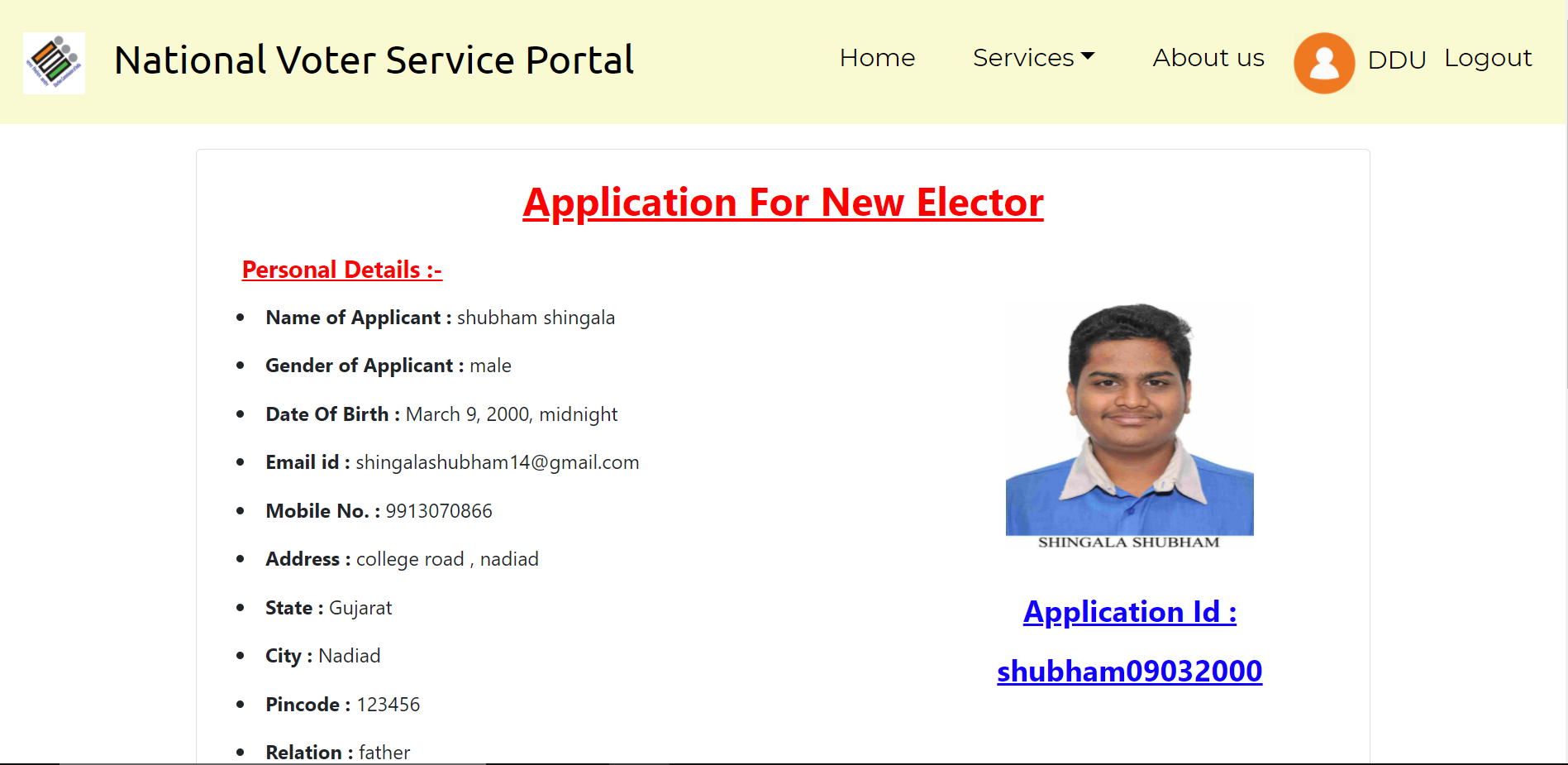


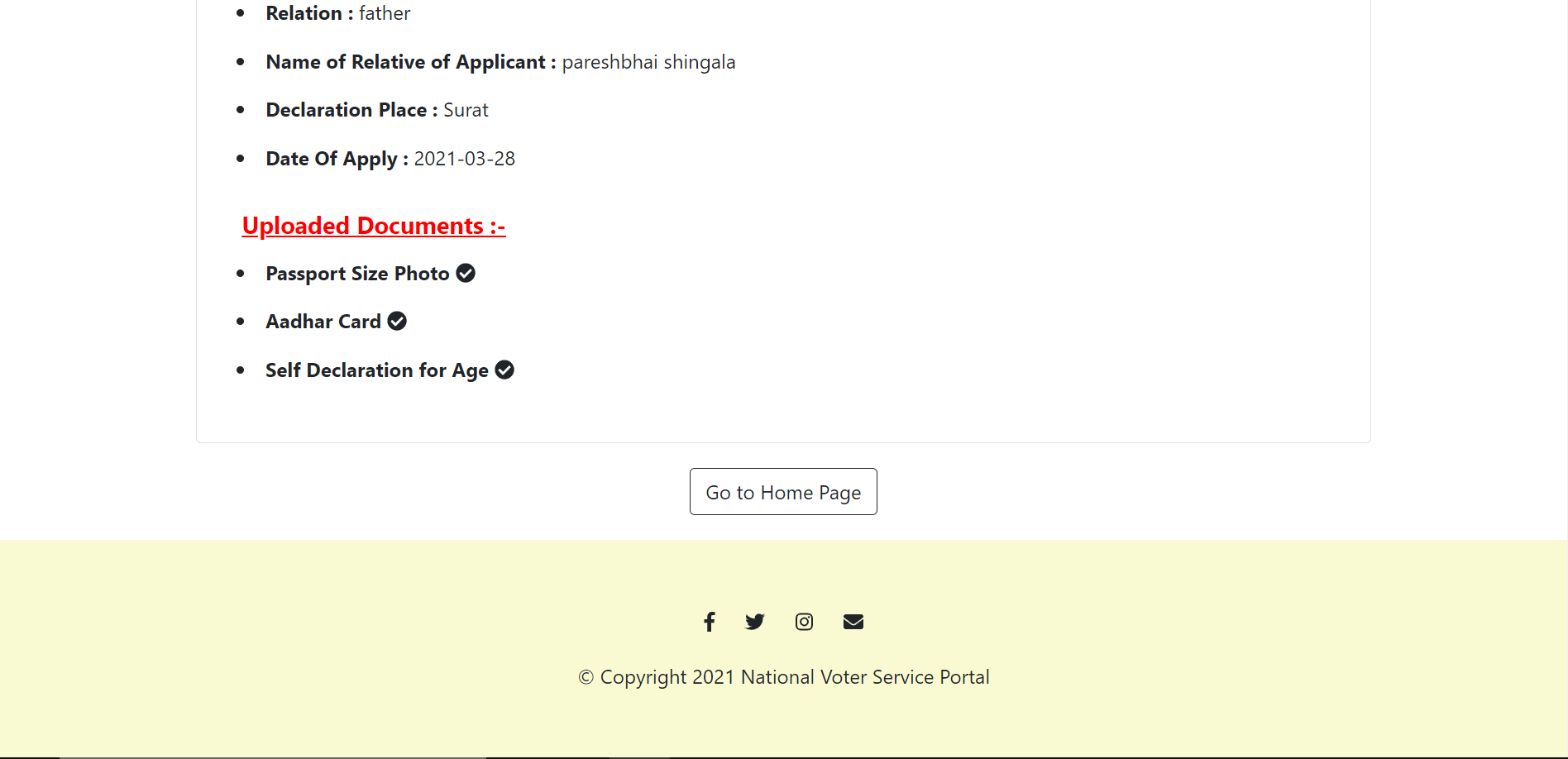


* Upload Documents :-

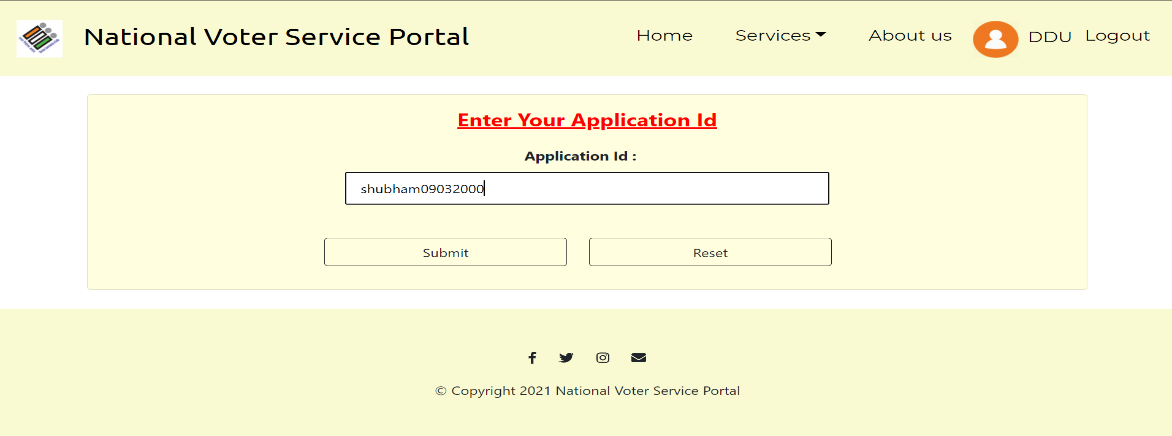


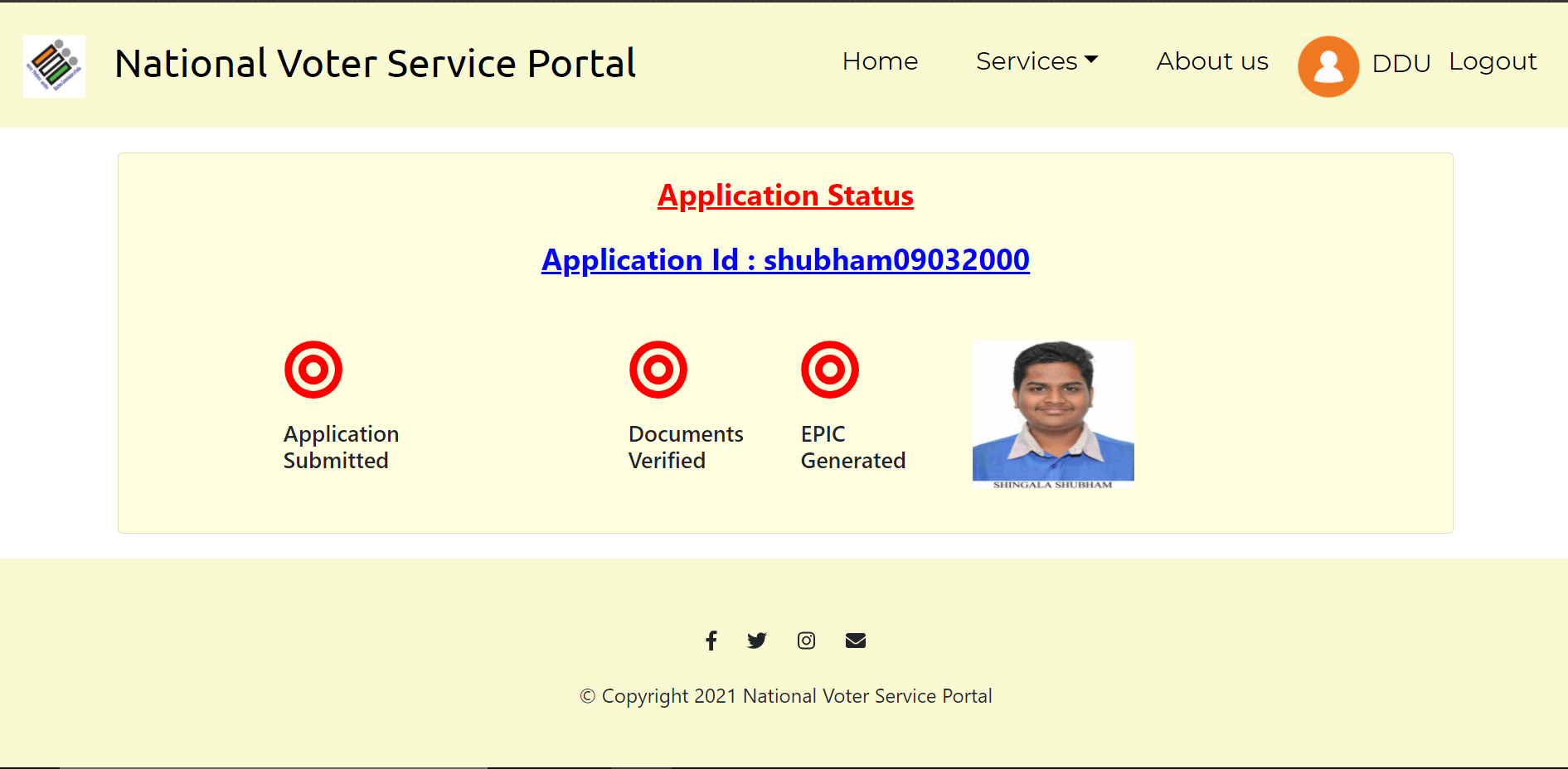
* Preview Application :-



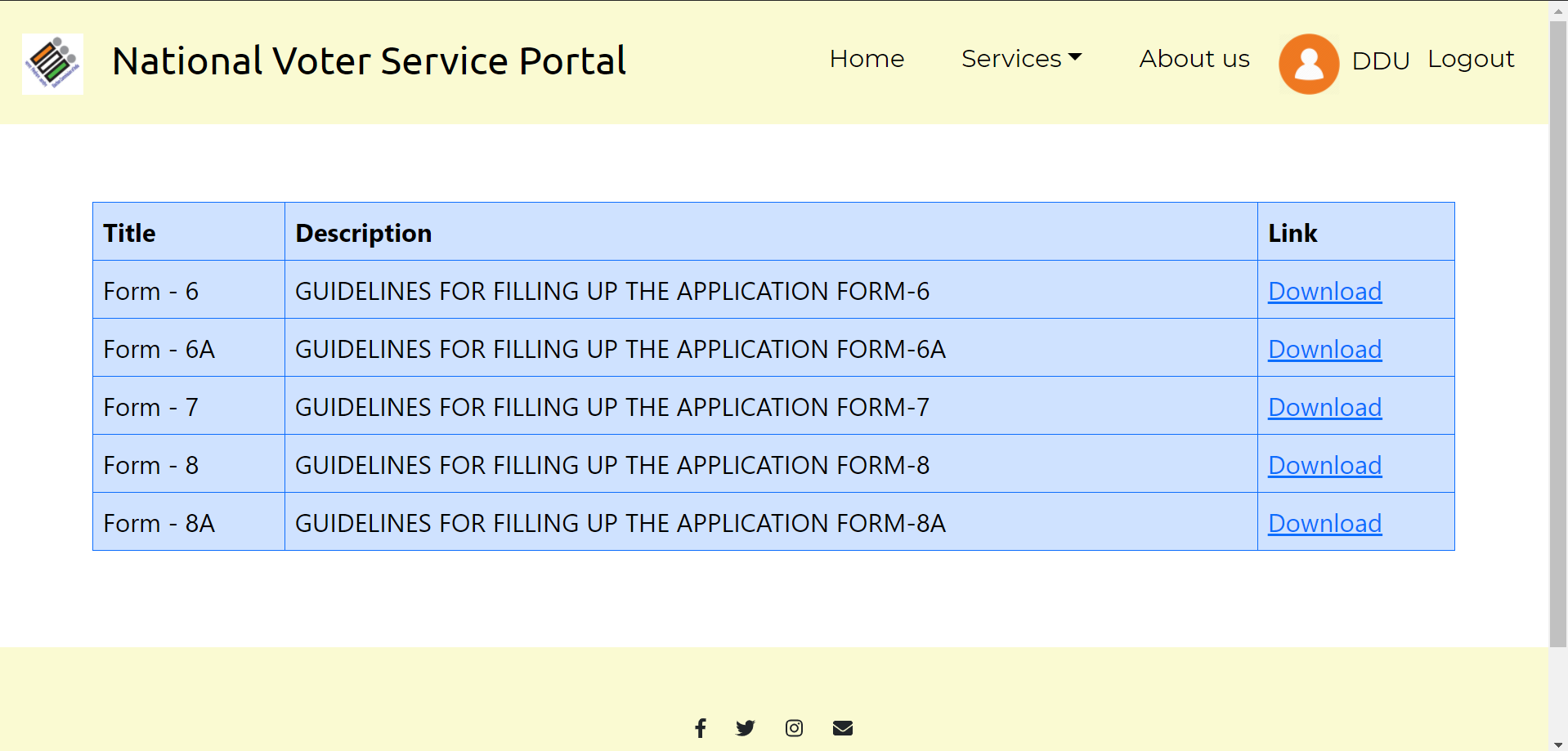


* Track Application Status :-





* Download Forms :-



1. **Conclusion**

The functionalities are implemented in system after understanding all the

System modules according to the requirements. Functionalities that are

Successfully implemented in the system are :

* User Registration
* Login
* User authentication
* Logout
* Apply for new voter id card
* Upload documents
* Preview & submit Application
* Track Application Status
* Download Forms
* Modification In voter id card
* Generate E-Epic
* Profile details

1. **Limitations and Future Enhancements**

We have implemented the functionality “Apply for new Voter id card”

Successfully with all its sub functionalities. The Project can be extended for

Some other functionalities like (1) “Download & Search Electoral Roll Pdf” as

per the selected State and city. (2) Verify Documents automatically at the time

of upload.

1. **Reference / Bibliography**

Following links and websites were referred during the development of this

Project:

docs.djangoproject.com

stackoverflow.com

github.com