Front - End Development

For

Vaccination Drives and Health Management System

Prepared By

Hiten Padaliya (201901401)
Satyam Bhut (201901408)
Dhairya Rupala (201901419)
Jenish Rathod (201901435)

Organization

Dhirubhai Ambani Institute of Information and Communication Technology

Backend Handling Program:

views.py

Vaccination > views.py

```python
from django.http.response import HttpResponse
from django.shortcuts import redirect, render
import psycopg2

# Setting up the connection to the Local database
connection = psycopg2.connect(host="localhost",database="201901419_SRS",user="postgres",password="admin",options='-c
cursor = connection.cursor()


# Method for Home page
def Home(request):
    return render(request,'Vaccination/index.html')

```

```python
# Method for Inventory handling
def addInventory(request):
    try:
        if request.method=="POST":
                if request.POST.get('i_insert'):
                    # Handling post request for insertion
                    i_ID = int(request.POST.get('i_ID'))
                    Name = "'"+request.POST.get('i_Name')+"'"
                    City = "'"+request.POST.get('i_City')+"'"
                    VaccineType = "'"+request.POST.get('i_type')+"'"
                    Stock = "'"+request.POST.get('i_stock')+"'"

                    querr = f'insert into inventory values({i_ID},{Name},{City},{VaccineType},{Stock})'
                    cursor.execute(querr)
                    connection.commit()

                    querr = 'select * from inventory'
                    cursor.execute(querr)
                    connection.commit()
                    tuples = cursor.fetchall()
                    context = {'tuples':tuples}
                    return render(request,'Vaccination/print_inventory.html',context)
                elif request.POST.get('i_search'):
                    # Handling post request for searching
                    querr = 'select * from inventory'
                    cursor.execute(querr)
                    connection.commit()
                    tuples = cursor.fetchall()
                    context = {'tuples':tuples}
                    return render(request,'Vaccination/print_inventory.html',context)
```

```
                elif request.POST.get('i_delete'):
                    # Handling post request for deletion
                    print('Delete func')
                    iid = request.POST.get('i_delete')
                    querr = f'delete from inventory where inventory_id={iid}'
                    cursor.execute(querr)
                    connection.commit()
                    print('Deleted successfully')
                    querr = 'select * from inventory'
                    cursor.execute(querr)
                    connection.commit()
                    tuples = cursor.fetchall()
                    context = {'tuples':tuples}
                    return render(request,'Vaccination/print_inventory.html',context)

    except:
        return HttpResponse('<p>Some Error Occured</p>')


    return render(request,'Vaccination/inventory.html')


```

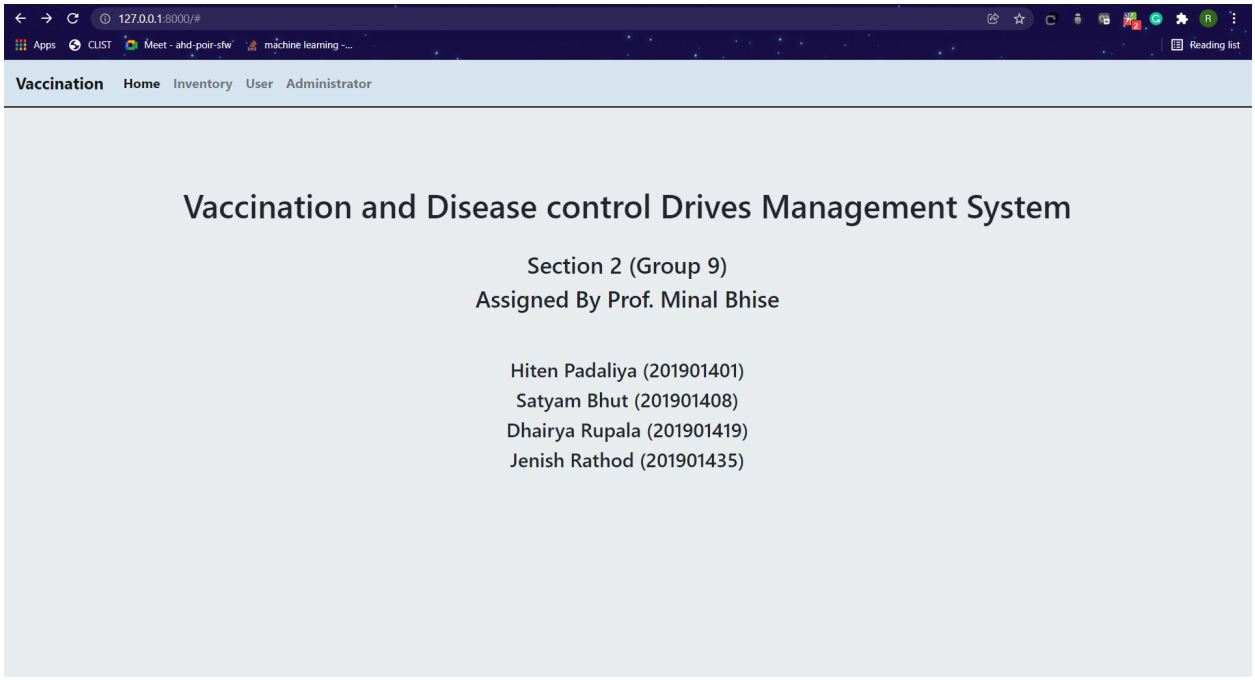
```
# Method for User handling
def addUser(request):
    try:
        if request.method=="POST":
                if request.POST.get('u_insert'):
                    # Handling post request for insertion
                    u_ID = int(request.POST.get('u_ID'))
                    u_CID = int(request.POST.get('u_Center_Id'))
                    u_SID = int(request.POST.get('u_Slot_ID'))
                    u_Name = "'"+request.POST.get('u_Name')+"'"
                    u_Gender = "'"+request.POST.get('u_Gender')+"'"
                    u_Age = int(request.POST.get('u_Age'))
                    u_Add = "'"+request.POST.get('u_Address')+"'"
                    u_City = "'"+request.POST.get('u_City')+"'"
                    u_VaccineType = "'"+request.POST.get('u_Vaccine')+"'"
                    u_IS = "'"+request.POST.get('u_Is_Vaccinated')+"'"

                    querr = f'insert into users values({u_ID},{u_CID},{u_SID},{u_Name},{u_Gender},{u_Age},{u_Add},{u_
                    cursor.execute(querr)
                    connection.commit()

                    querr = 'select * from users'
                    cursor.execute(querr)
                    connection.commit()
                    tuples = cursor.fetchall()
                    context = {'tuples':tuples}
                    return render(request,'Vaccination/print_user.html',context)
```

```
93                  elif request.POST.get('u_search'):
94                      # Handling post request for searching
95                      querr = 'select * from users'
96                      cursor.execute(querr)
97                      connection.commit()
98                      tuples = cursor.fetchall()
99                      context = {'tuples':tuples}
100                     return render(request,'Vaccination/print_user.html',context)
101                 elif request.POST.get('u_delete'):
102                     # Handling post request for deletion
103                     u_id = request.POST.get('u_delete')
104                     querr = f'delete from users where user_id={u_id}'
105                     cursor.execute(querr)
106                     connection.commit()
107
108                     querr = 'select * from users'
109                     cursor.execute(querr)
110                     connection.commit()
111                     tuples = cursor.fetchall()
112                     context = {'tuples':tuples}
113                     return render(request,'Vaccination/print_user.html',context)
114     except:
115         return HttpResponse('<p> Some error occured</p>')
116
117     return render(request,'Vaccination/user.html')
118
119
```

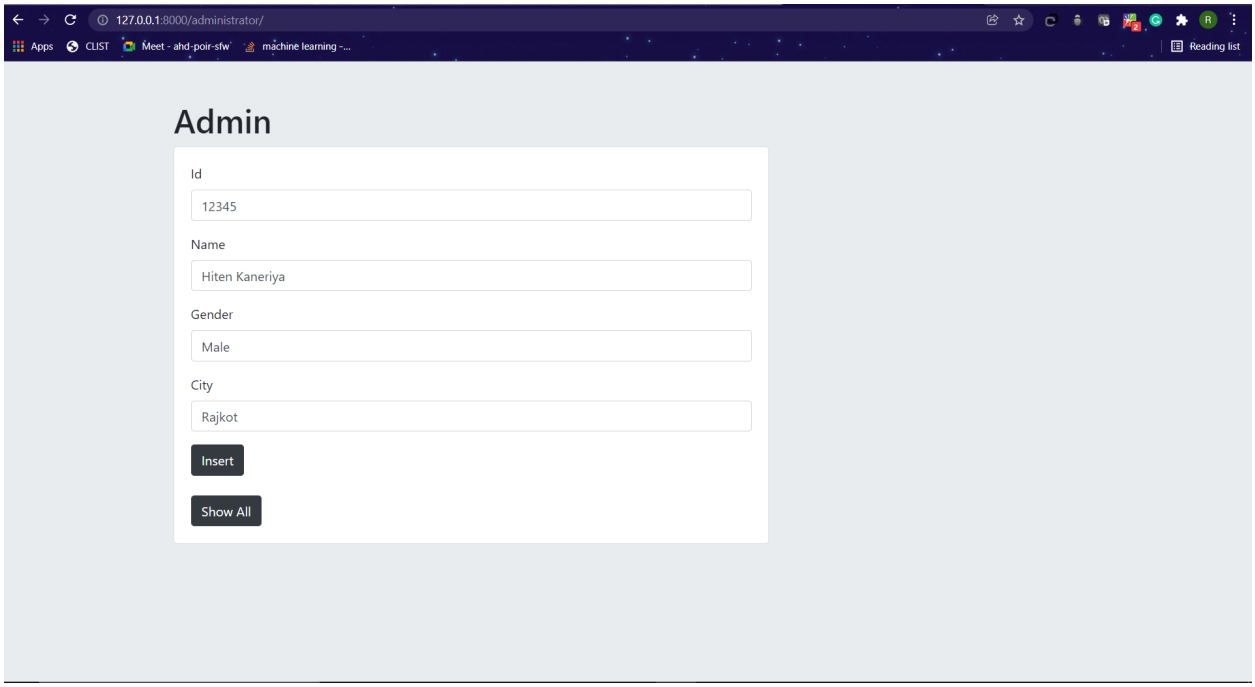
```
120 # Method for Admin handling
121 def addAdmin(request):
122     try:
123         if request.method=="POST":
124                 if request.POST.get('a_insert'):
125                     # Handling post request for insertion
126                     a_ID = int(request.POST.get('a_ID'))
127                     Name = "'"+request.POST.get('a_Name')+"'"
128                     Gender = "'"+request.POST.get('a_Gender')+"'"
129                     City = "'"+request.POST.get('a_City')+"'"
130                     querr = f'insert into administrator values({a_ID},{Name},{Gender},{City})'
131                     cursor.execute(querr)
132                     connection.commit()
133
134                     querr = 'select * from administrator'
135                     cursor.execute(querr)
136                     connection.commit()
137                     tuples = cursor.fetchall()
138                     context = {'tuples':tuples}
139                     return render(request,'Vaccination/print_admin.html',context)
140                 elif request.POST.get('a_search'):
141                     # Handling post request for searching
142                     querr = 'select * from administrator'
143                     cursor.execute(querr)
144                     connection.commit()
145                     tuples = cursor.fetchall()
146                     context = {'tuples':tuples}
147                     return render(request,'Vaccination/print_admin.html',context)
```

```
148                 elif request.POST.get('a_delete'):
149                     # Handling post request for deletion
150                     a_id = request.POST.get('a_delete')
151                     querr = f'delete from administrator where admin_id={a_id}'
152                     cursor.execute(querr)
153                     connection.commit()
154 
155                     querr = 'select * from administrator'
156                     cursor.execute(querr)
157                     connection.commit()
158                     tuples = cursor.fetchall()
159                     context = {'tuples':tuples}
160                     return render(request,'Vaccination/print_admin.html',context)
161         except:
162             return HttpResponse('<p> Some error occured</p>')
163 
164         return render(request,'Vaccination/admin.html')
165 
```

Home Page of the Website:



Admin Form of the Website:



Showing the Admin Data: