**OVERVIEW**

**Submitted by,**

**Neena K**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **CONTENT** | **PAGE NO** |
| **1** | Program to Add two Numbers | **5** |
| **2** | Program to check Armstrong number | **6** |
| **3** | Find the number of elements in the array | **7** |
| **4** | To sort the elements in the array | **8** |
| **5** | To find the Fibonacci series | **9** |
| **6** | To split the array | **11** |
| **7** | Program to find the matrix multiplication | **12** |
| **8** | To find multiplication table of a number | **13** |
| **9** | To create simple login and logout | **14** |
| **10** | Simple registration form | **16** |
| **11** | Internspace | **19** |
| **12** | Abstract | **20** |
| **13** | Code | **21** |
| **14** | Screenshots | **49** |

**INDEX**

**Django**

Django is a Python-based free and open-source web framework that follows the model-view-controller (MVC) architectural pattern.[5][6] It is maintained by the Django Software Foundation (DSF), an American independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself.[7] Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

**Python**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

**SQLLite**

SQLLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

SQLite is ACID-compliant and implements most of the SQL standard, generally following PostgreSQL syntax. However, SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.[7] This means that one can, for example, insert a string into a column defined as an integer. SQLite will attempt to convert data between formats where appropriate, the string "123" into an integer in this case, but does not guarantee such conversions and will store the data as-is if such a conversion is not possible.

SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others.[8] SQLite has bindings to many programming languages.

**PYTHON PROGRAMS**

**1.Program to add two numbers.**

a = int(input("enter first number: "))

b = int(input("enter second number: "))

sum = a + b

print("sum:", sum)

**OUTPUT**

enter first number:5

enter second number:7

sum: 12

**2. Program to accept a number and check whether it is amstrong or not.**

num = int(input("Enter a number: "))

sum = 0

temp = num

while temp > 0:

digit = temp % 10

sum += digit \*\* 3

temp //= 10

if num == sum:

print(num,"is an Armstrong number")

else:

print(num,"is not an Armstrong number")

**OUTPUT**

Enter a number: 663

663 is not an Armstrong number

**3.**[**Program to find number of elements in an array.**](http://www.makeitsimple.co.in/PHP_countelements.php)

num = int(input("Enter a number: "))

if (num % 2) == 0:

print("{0} is Even".format(num))

else:

print("{0} is Odd".format(num))

**OUTPUT**

Enter a number: 43

43 is Odd

**4.** [**Program to sort elements in an array in ascending order.**](http://www.makeitsimple.co.in/PHP_sortarray.php)

arr = [5, 2, 8, 7, 1];

temp = 0;

print("Elements of original array: ");

for i in range(0, len(arr)):

print(arr[i]);

for i in range(0, len(arr)):

for j in range(i+1, len(arr)):

if(arr[i] > arr[j]):

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

print();

print("Elements of array sorted in ascending order: ");

for i in range(0, len(arr)):

print(arr[i]);

**OUTPUT**

Elements of original array:

5 2 8 7 1

Elements of array sorted in ascending order:

1 2 5 7 8

**5.** [**Program to find Fibonacci series.**](http://www.makeitsimple.co.in/PHP_arraysum.php)

nterms = int(input("How many terms? "))

# first two terms

n1, n2 = 0, 1

count = 0

# check if the number of terms is valid

if nterms <= 0:

print("Please enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence upto",nterms,":")

print(n1)

else:

print("Fibonacci sequence:")

while count < nterms:

print(n1)

nth = n1 + n2

# update values

n1 = n2

n2 = nth

count += 1

**OUTPUT**

How many terms? 7

Fibonacci sequence:

0

1

1

2

3

5

8

**6.** [**Program to split a string as array elements based on delimiter.**](http://www.makeitsimple.co.in/PHP_splitarray.php)

def split\_string(string):

list\_string = string.split(' ')

return list\_string

def join\_string(list\_string):

string = '-'.join(list\_string)

return string

if \_\_name\_\_ == '\_\_main\_\_':

string = ' God’s own country '

list\_string = split\_string(string)

print(list\_string)

new\_string = join\_string(list\_string)

print(new\_string)

**OUTPUT**

Input : God’s own country

Output : ['God’s', own, 'country']

**7. Program to find the multiplication of the matrix**

X = [[12,7,3],

[4 ,5,6],

[7 ,8,9]]

Y = [[5,8,1],

[6,7,3],

[4,5,9]]

result = [[0,0,0],

[0,0,0],

[0,0,0]]

for i in range(len(X)):

result[i][j] = X[i][j] + Y[i][j]

for r in result:

print(r)

**OUTPUT**

[114, 160, 60, 27]

[74, 97, 73, 14]

[119, 157, 112, 23]

**8.** **Program to find the multiplication table of a number**

num = int(input("Show multiplication table of ?"))

for i in range(1,11):

print(num,'x',i,'=',num\*i)

**OUTPUT**

Show multiplication table of ?

12

12 x 1 = 12

12 x 2 = 24

12 x 3 = 36

12 x 4 = 48

12 x 5 = 60

12 x 6 = 72

12 x 7 = 84

12 x 8 = 96

12 x 9 = 108

12 x 10 = 120

**9.** [**Programs to create simple Login and Logout .**](http://www.makeitsimple.co.in/PHP_loginexmp.php)

**LOGIN**

**urls.py**

from django.urls import path

from . import views

app\_name = 'main' # here for namespacing of urls.

urlpatterns = [

path("", views.homepage, name="homepage"),

path("register/", views.register, name="register"),

path("logout", views.logout\_request, name="logout"),

]

**views.py**

def logout\_request(request):

logout(request)

messages.info(request, "Logged out successfully!")

return redirect("main:homepage")

**login.html**

{% extends 'main/header.html' %}

{% block content %}

<div class="container">

<form method="POST">

{% csrf\_token %}

{{form.as\_p}}

<button style="background-color:#F4EB16; color:blue" class="btn btn-outline-info" type="submit">Login</button>

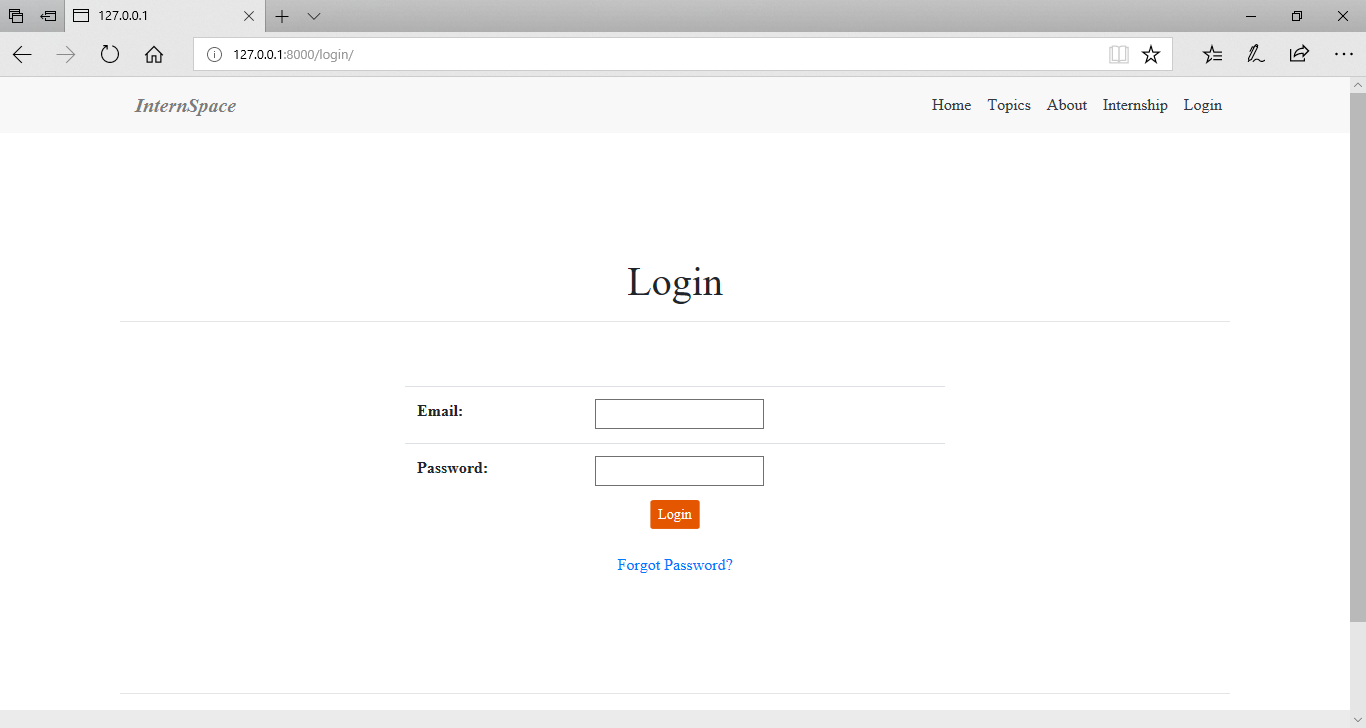
</form>

Don't have an account? <a href="/register" target="blank"><strong>register here</strong></a>!

</div>

{% endblock %}

**OUTPUT**

****

**10.** [**Program to create a simple Registration form.**](http://www.makeitsimple.co.in/PHP_registration.php)

**forms.py**

from django import forms

from django.contrib.auth.forms import UserCreationForm

from django.contrib.auth.models import User

class SignUpForm(UserCreationForm):

first\_name = forms.CharField(max\_length=30, required=False, help\_text='Optional.')

last\_name = forms.CharField(max\_length=30, required=False, help\_text='Optional.')

email = forms.EmailField(max\_length=254, help\_text='Required. Inform a valid email address.')

class Meta:

model = User

fields = ('username', 'first\_name', 'last\_name', 'email', 'password1', 'password2', )

**views.py**

from django.contrib.auth import login, authenticate

from django.shortcuts import render, redirect

from mysite.core.forms import SignUpForm

def signup(request):

if request.method == 'POST':

form = SignUpForm(request.POST)

if form.is\_valid():

form.save()

username = form.cleaned\_data.get('username')

raw\_password = form.cleaned\_data.get('password1')

user = authenticate(username=username, password=raw\_password)

login(request, user)

return redirect('home')

else:

form = SignUpForm()

return render(request, 'signup.html', {'form': form})

**models.py**

from django.db import models

from django.contrib.auth.models import User

from django.db.models.signals import post\_save

from django.dispatch import receiver

class Profile(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

bio = models.TextField(max\_length=500, blank=True)

location = models.CharField(max\_length=30, blank=True)

birth\_date = models.DateField(null=True, blank=True)

@receiver(post\_save, sender=User)

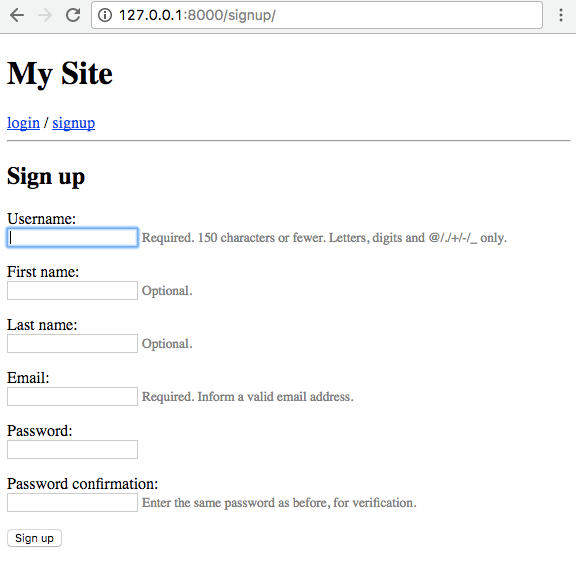
def update\_user\_profile(sender, instance, created, \*\*kwargs):

if created:

Profile.objects.create(user=instance)

instance.profile.save()

**OUTPUT**



Internspace

**ABSTRACT**

Internspace is a content management system. It can be used for doing internship. This portal provides a facility for accessing the internship by the intern. Intern can login to the website and they can search and apply for the available internships. Math Internships and Workshops (on Visualizing Math with Open-Source toolkits) are conducted throughout the year. Visit our Outreach page for more details on the same.

Internships are from selected topics in Mathematics, like Real Analysis, Linear Algebra and Calculus. Our focus lies primarily on an undergraduate level of Mathematics. We also conduct workshops in colleges and universities across India.

The main users in the project can be categorized into three modules Admin, Interns and Staff. Admin can manage the Staffs and can view the applied Interns. Admin can add/delete/ approve the interns who is registered. The Staff can create internship, view the internships and also view the application by the interns..

**CODE**

**settings.py**

import os

# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)

import tempfile

# email config

from .email\_config import (

                    EMAIL\_HOST,

                    EMAIL\_PORT,

                    EMAIL\_HOST\_USER,

                    EMAIL\_HOST\_PASSWORD,

                    EMAIL\_USE\_TLS,

                    SENDER\_EMAIL,

                    SECRET\_KEY\_SETTINGS

                    )

BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = '%ses\_=j4dd31z!k-tnszg+gravu0v-vtn4\*#6i8z5n2buje1rk'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = ["\*"]

# Application definition

INSTALLED\_APPS = [

    'django.contrib.humanize',

    # 'phonenumber\_field',

    'fossee\_math\_pages.apps.FosseeMathPagesConfig',

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'ckeditor',

    'ckeditor\_uploader',

    'crispy\_forms',

    'django\_bootstrap\_breadcrumbs',

]

CRISPY\_TEMPLATE\_PACK = 'bootstrap4'

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'FOSSEE\_math.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [os.path.join(BASE\_DIR, 'templates')],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

BREADCRUMBS\_TEMPLATE = os.path.join(BASE\_DIR, 'templates/fossee\_math\_pages/breadcrumbs\_template.html')

WSGI\_APPLICATION = 'FOSSEE\_math.wsgi.application'

# Database

# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

    }

}

# Password validation

# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

# Internationalization

# https://docs.djangoproject.com/en/2.2/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'Asia/Kolkata'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

LOGOUT\_REDIRECT\_URL = 'index'

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/2.2/howto/static-files/

STATIC\_ROOT = os.path.join(BASE\_DIR, 'static/')

STATIC\_URL = '/static/'

STATICFILES\_DIRS = (

    os.path.join(BASE\_DIR, 'FOSSEE\_math/static'),

)

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

CKEDITOR\_UPLOAD\_PATH = "uploads/"

CKEDITOR\_CONFIGS = {

    'default': {

        'skin': 'moono-lisa',

        'toolbar\_Basic': [

            ['Source', '-', 'Bold', 'Italic']

        ],

        'toolbar\_YourCustomToolbarConfig': [

            {'name': 'document', 'items': ['Source', '-', 'Preview']},

            {'name': 'clipboard', 'items': ['Cut', 'Copy', 'Paste', 'PasteText', 'PasteFromWord', '-', 'Undo', 'Redo']},

            {'name': 'editing', 'items': ['Find', 'Replace', '-', 'SelectAll']},

            {'name': 'insert', 'items': ['Table', 'HorizontalRule', 'Smiley', 'SpecialChar', 'PageBreak']},

            {'name': 'basicstyles',

             'items': ['Bold', 'Italic', 'Underline', 'Strike', 'Subscript', 'Superscript', '-', 'RemoveFormat']},

            {'name': 'tools', 'items': ['Maximize', 'ShowBlocks']},

            {'name': 'paragraph',

             'items': ['NumberedList', 'BulletedList', '-', 'Outdent', 'Indent', '-', 'Blockquote', 'CreateDiv', '-',

                       'JustifyLeft', 'JustifyCenter', 'JustifyRight', 'JustifyBlock', '-', 'BidiLtr', 'BidiRtl']},

            {'name': 'styles', 'items': ['Styles', 'Format', 'Font', 'FontSize']},

            {'name': 'colors', 'items': ['TextColor', 'BGColor']},

            {'name': 'math', 'items': ['Mathjax', ]},

        ],

        'toolbar': 'YourCustomToolbarConfig',  # put selected toolbar config here

        'toolbarGroups': [{'name': 'document', 'groups': ['mode', 'document', 'doctools']}],

        'height': 500,

        'width': 1000,

        'filebrowserWindowHeight': 725,

        'filebrowserWindowWidth': 940,

        'toolbarCanCollapse': True,

        'tabSpaces': 4

        'mathJaxLib': '//cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS\_HTML',

        'extraPlugins': ','.join(['mathjax', ]),

    }

}

# messages

from django.contrib.messages import constants as messages

MESSAGE\_TAGS = {

    messages.ERROR: 'danger'

}

if DEBUG:

    EMAIL\_BACKEND = 'django.core.mail.backends.console.EmailBackend'

# Email config

EMAIL\_HOST = EMAIL\_HOST

EMAIL\_HOST\_USER = EMAIL\_HOST\_USER

EMAIL\_HOST\_PASSWORD = EMAIL\_HOST\_PASSWORD

EMAIL\_PORT = EMAIL\_PORT

EMAIL\_USE\_TLS = EMAIL\_USE\_TLS

EMAIL\_TIMEOUT = 300

SENDER\_EMAIL = SENDER\_EMAIL

EMAIL\_BACKEND = 'django.core.mail.backends.console.EmailBackend'

# EMAIL\_USE\_TLS = True

**urls.py**

from django.conf import settings

from django.conf.urls import include

from django.conf.urls.static import static

from django.contrib import admin

from django.contrib.auth import views as auth\_views

from django.urls import path

import fossee\_math\_pages.views as myview

urlpatterns = [

    path('', include('fossee\_math\_pages.urls')),

    path('admin/', admin.site.urls),

    path('ckeditor/', include('ckeditor\_uploader.urls')),

    path('password\_reset/done/',

         auth\_views.PasswordResetCompleteView.as\_view(template\_name='password\_reset/password\_reset\_done.html'),

         name='password\_reset\_done'),

    path('reset/<uidb64>/<token>/',

         auth\_views.PasswordResetConfirmView.as\_view(template\_name='password\_reset/password\_reset\_confirm.html'),

         name='password\_reset\_confirm'),

    path('password\_reset/',

         auth\_views.PasswordResetView.as\_view(template\_name='password\_reset/password\_reset\_form.html'),

         name='password\_reset'),

    path('reset/done/',

         auth\_views.PasswordResetCompleteView.as\_view(template\_name='password\_reset/password\_reset\_complete.html'),

         name='password\_reset\_complete'),

    path('activate/<uidb64>/<token>/',

         myview.activate, name='activate'),

]

urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

handler404 = 'fossee\_math\_pages.views.error\_404\_view'

**wsgi.py**

import os

from django.core.wsgi import get\_wsgi\_application

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'FOSSEE\_math.settings')

application = get\_wsgi\_application()

**email\_config.py**

EMAIL\_HOST = 'your smtp host name'

EMAIL\_PORT = 'PORT Number'

EMAIL\_HOST\_USER = 'your username'

EMAIL\_HOST\_PASSWORD = 'your password'

EMAIL\_USE\_TLS = True

SENDER\_EMAIL = 'your email address'

SECRET\_KEY\_SETTINGS = ‘ ‘

**index.html**

{% extends 'base.html' %}{% load static %}{% block title %}Home{% endblock %}{% block content %}

    <!-- Page Content -->

    <div class="container">

        <br><br>

        <!-- Heading Row -->

        <div class="row align-items-center my-5">

            <!-- /.col-lg-8 -->

            <div class="col-lg-5">

                <h1 class="font-weight-light serif">InternSpace</h1>

                <p class="serif">Make complex mathematical topics feel more natural and approachable</p>

            </div>

            <div class="col-lg-7">

                <img class="img-fluid rounded mb-4 mb-lg-0" src="{% static 'web-files/img/index-images/logo2.png' %}" alt="">

            </div>

            <!-- /.col-md-4 -->

        </div>

        <!-- /.row -->

        <br>

    </div>

    <hr>

    <div class="container" id="topics">

        <div>

            <form method="GET" action=".">

                <table>

                    <tr>

                        <td> Search :</td>

                        <td>

                            <input class="form-control py-2 border" type="search"

                                   name="title\_contains"

                                   placeholder="Search For Topics or Sub-topics"/>

                        </td>

                        <td>

                            <button type="submit" class="btn btn-primary">Search</button>

                        </td>

                    </tr>

                </table>

            </form>

        </div>

<!--        <hr style="background-color: #902b2b;height: 1px">-->

    </div>

    <hr>

    <div class="container">

        <h1>Topics</h1>

        <hr> {% for i in internship %}

        <div class="card mb-3 col-md-12">

            <div class="row row-no-gutters">

                <div class="col-md-2">

                    <img class="card-img" src="{{ i.internship\_thumbnail.url }}" alt="">

                </div>

                <div class="col-md-10">

                    <div class="card-body">

                        <h5 class="card-title serif">

                            <b><a href="{% url 'contents' i.internship\_url %}">{{ i.internship\_topic }}</a></b></h5>

                        <span class="card-text serif"> <i>"{{ i.internship\_quote }}"</i></span>

                        <p class="card-text">

                            <small class="text-muted">

                                -- {{ i.internship\_quote\_author }}</small>

                        </p>

                    </div>

                </div>

            </div>

        </div>

    {% endfor %}

        <br>

    </div>

{% endblock %}

**interns.html**

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} View Interns {% endblock %}

{% load django\_bootstrap\_breadcrumbs %}

{% block breadcrumbs %}

    {{ block.super }}

    {% breadcrumb "Dashboard" "dashboard" %}

    {% breadcrumb "Interns" "interns" %}

{% endblock %}

{% block content %}

    {% render\_breadcrumbs %}

    <div class="container">

        <h3 class="text-warning">Overview</h3>

        <form method="POST">

            {% csrf\_token %}

            <label>Select Internship: </label>

            <select name="search\_internship" onchange="submit();">

                {% if internship in internship\_all %}

                    <option>{{ internship }}</option>

                {% else %}

                    <option>Select Internship</option>

                {% endif %}

                {% for i in internship\_all %}

                    {% if i != internship %}

                        <option value="{{ i.pk }}">{{ i.internship\_topic }}</option>

                    {% endif %}

                {% endfor %}

            </select>

        </form>

        <br/>

        <table class="table table-light table-sm">

            <thead class="table-warning">

            <tr>

                <th>Intern Name</th>

                <th>Internship</th>

                <th>Topic</th>

                <th>Subtopic</th>

            </tr>

            </thead>

            <tbody>

            {% for i in topics %}

                {% if internship.pk == i.topic\_id.internship\_id\_id %}

                    <tr>

                        {% if i.assigned\_user\_id\_id != None %}

                            <td>{{ i.assigned\_user\_id }}</td>

                        {% else %}

                            <td></td>

                        {% endif %}

                        <td>{{ i.topic\_id.internship\_id }}</td>

                        <td>{{ i.topic\_id }}</td>

                        <td>{{ i.subtopic\_name }}</td>

                    </tr>

                {% endif %}

            {% endfor %}

            </tbody>

        </table>

    </div>

    <br>

    <hr>

    <br> {% endblock %}

**login.html**

{% extends 'base.html' %} {% block title %}Login{% endblock %} {% block content %}

    <div class="container" align="justify" style="margin-top:100px;margin-bottom:100px;">

        <br> {% include 'partials/\_alerts.html' %}

        <center>

            <h1>Login</h1>

            <hr>

            <br>

            <br>

            <div class="container" style="font-family: 'Lato', sans-serif;">

                <form method="post">

                    {% csrf\_token %}

                    <table class="table" style="margin: auto; width: 50%;">

                        {{ form.as\_table }}

                    </table>

                    <button class="btn btn-success btn-sm" type="submit"

                            style="background-color:#e45700;color:white; border:0px">Login

                    </button>

                    <br>

                    <br>

                    <a href="{% url 'password\_reset' %}">Forgot Password?</a>

                </form>

            </div>

        </center>

    </div> {% endblock %}

**messages.html**

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% load static %}

{% block title %} Add Submission {% endblock %}

{% load template\_vars %}

{% load django\_bootstrap\_breadcrumbs %}

{% block breadcrumbs %}

{{ block.super }}

{% breadcrumb "Dashboard" "dashboard" %}

{% if not user.is\_staff %}

    {% breadcrumb "Internship" "add-submission" %}

{% else %}

    {% breadcrumb "Review Submissions" "review-submissions" %}

{% endif %}

{% breadcrumb "Messages" "message" %}

<!-- WRONG CRUMBS -->

{% endblock %}

{% block content %}

{% render\_breadcrumbs %}

{% if subtopic.subtopic\_status == "ACCEPTED" %}

    {% setvar "success" as color %}

{% elif subtopic.subtopic\_status == "WAITING" %}

    {% setvar "warning" as color %}

{% elif subtopic.subtopic\_status == "REJECTED" %}

    {% setvar "danger" as color%}

{% endif %}

<div class="container">

    <div class="card-header bg-{{ color }} rounded-top">

        <!-- if subtopic status -->

        <h3 class="text-white"><u>{{ subtopic.subtopic\_name }}</u>

        <img src="{% static '/web-files/img/multiuse-svgs/comment-discussion-white.svg' %} " onload=scrollToBottom()>

        </h3>

        <h6 class="text-light">{{ subtopic.topic\_id.internship\_id }} &rarr; {{ subtopic.topic\_id }}

            &rarr; {{ subtopic.subtopic\_name }}</h6>

    </div>

    <br />

    <hr />

{% if user.is\_staff %}

    <div class="row">

        <div class="col-md-10  offset-md-1">

            <div class="border" style="height: 600px; overflow-y: auto;" id="table-chat" onclick="scrollToBottom()">

                <table class="table table-borderless table-sm" id="chat-table">

                    <thead>

                        <tr></tr>

                    </thead>

                    <tbody>

                        {% for i in message %}

                        {% if i.subtopic\_id\_id == subtopic.pk %}

                        <tr>

                            <td>

                                {% if i.user\_id.is\_staff %}

                                   <span class="text-muted" style="font-size: smaller;"><p  class="text-right">Sent on : {{ i.message\_send\_date }} by <b>{{ i.user\_id }}</b></p></span>

                                    <div class="bs-callout-right bs-callout-right-info text-right" style="margin-top:0px; width: 100%; max-width: 900px ;"><p>{{ i.message }}</p></div>

                                {% elif not i.user\_id.is\_staff %}

                                <span class="text-muted" style="font-size: smaller;"><p>Sent on : {{ i.message\_send\_date }} by <b>{{ i.user\_id }}</b></p></span>

                                <div class="bs-callout bs-callout-{{ color }}" style="margin-top: 0px;"><p>{{ i.message }}</p></div>

                                {% endif %}

                            </td>

                        </tr>

                        {% endif %}

                        {% endfor %}

                    </tbody>

                </table>

            </div>

            <hr/>

        </div>

    </div>

{% elif not user.is\_staff and not user.is\_superuser %}

<div class="row">

    <div class="col-md-10  offset-md-1">

        <div class="border" style="height: 600px; overflow-y: auto;" id="table-chat" onclick="">

            <table class="table table-borderless table-sm" id="chat-table" style="width: 100%; max-width: 900px ">

                <thead>

                    <tr></tr>

                </thead>

                <tbody>

                    {% for i in message %}

                    {% if i.subtopic\_id\_id == subtopic.pk %}

                    <tr>

                        <td>

                            {% if i.user\_id.is\_staff %}

                               <span class="text-muted" style="font-size: smaller;"><p>Sent on : {{ i.message\_send\_date }} by <b>{{ i.user\_id }}</b></p></span>

                                <div class="bs-callout bs-callout-{{ color }}" style="margin-top:0px; width: 100%; max-width: 900px ;"><p>{{ i.message }}</p></div>

                            {% elif not i.user\_id.is\_staff %}

                            <span class="text-muted text-right" style="font-size: smaller;"><p class="text-right">Sent on : {{ i.message\_send\_date }} by <b>{{ i.user\_id }}</b></p></span>

                            <div class="bs-callout-right bs-callout-right-info text-right" style="margin-top: 0px;"><p class="text-right">{{ i.message }}</p></div>

                            {% endif %}

                        </td>

                    </tr>

                    {% endif %}

                    {% endfor %}

                </tbody>

            </table>

        </div>

        <hr/>

    </div>

</div>

{% endif %}

        <form action="javascript:addData()" method="POST">

            <div class="input-group offset-md-1">

                {% csrf\_token %}{{ form|crispy }}

                <div class="input-group-append">

                    <button class="btn btn-light" onClick="event.preventDefault();addData();">

                        <span class=""><img src="{% static '/web-files/img/multiuse-svgs/send.svg' %}"

                                width=18px></span>

                        Send</button>

                </div>

            </div>

            <!--<input class="btn btn-outline-success btn-md btn-block" type="button" onClick="addData()" id="submitAJAX" value="Add Data">-->

        </form>

  <hr>

</div>

<br>

<script>

    window.onload = scrollToBottom;

    function addData(){

            var data = document.getElementById("id\_message").value;

            console.log(data)

            var xhttp = new XMLHttpRequest();

            var csrf\_token = '{{ csrf\_token }}';

            if (data == null || data.trim() === '') {

            } else {

                console.log(data)

                var mainTable = document.getElementById("chat-table");

                var row = mainTable.insertRow()

                var cell1 = row.insertCell(0)

                cell1.innerHTML = "<div class=\"bs-callout-right bs-callout-right-default text-right text-muted\" style=\"margin-top: 0px;\"><p class=\"text-right\">"+data+"</p></div>";

                scrollToBottom()

                xhttp.onreadystatechange = function () {

                    if (this.readyState == 4 && this.status == 200) {

                        cell1.innerHTML = "<div class=\"bs-callout-right bs-callout-right-info text-right \" style=\"margin-top: 0px;\"><p class=\"text-right\">"+data+"</p></div>";

                        document.getElementById("id\_message").value = "";

                    }

                };

                xhttp.open('post', "/dashboard/messages/{{ subtopic.subtopic\_hash }}", true);

                xhttp.setRequestHeader("X-CSRFToken", csrf\_token);

                xhttp.setRequestHeader('Content-type', "application/x-www-form-urlencoded");

                xhttp.send("message=" + encodeURIComponent(data));

            }

        }

        function scrollToBottom() {

        var element = document.getElementById("table-chat");

        //window.scrollTo(0, element.offsetHeight);

        element.scrollTop = element.scrollHeight;

    }

</script>

{% endblock %}

**change\_password.html**

{% extends 'base.html' %} {% block title %}Password Change{% endblock %} {% block content %}

    <div class="container" align="justify" style="margin-top:100px;margin-bottom:100px;">

        <br> {% include 'partials/\_alerts.html' %}

        <center>

            <h1>Change Password</h1>

            <hr>

            We recomend you to change your password on first login !

            <br>

            <br>

            <div class="container" style="font-family: 'Lato', sans-serif;">

                <form method="post">

                    {% csrf\_token %}

                    <table class="table" style="margin: auto; width: 50%;">

                        {{ form.as\_table }}

                    </table>

                    <button class="btn " type="submit"

                            style="background-color:#e45700;color:white;">Set Password

                    </button>

                </form>

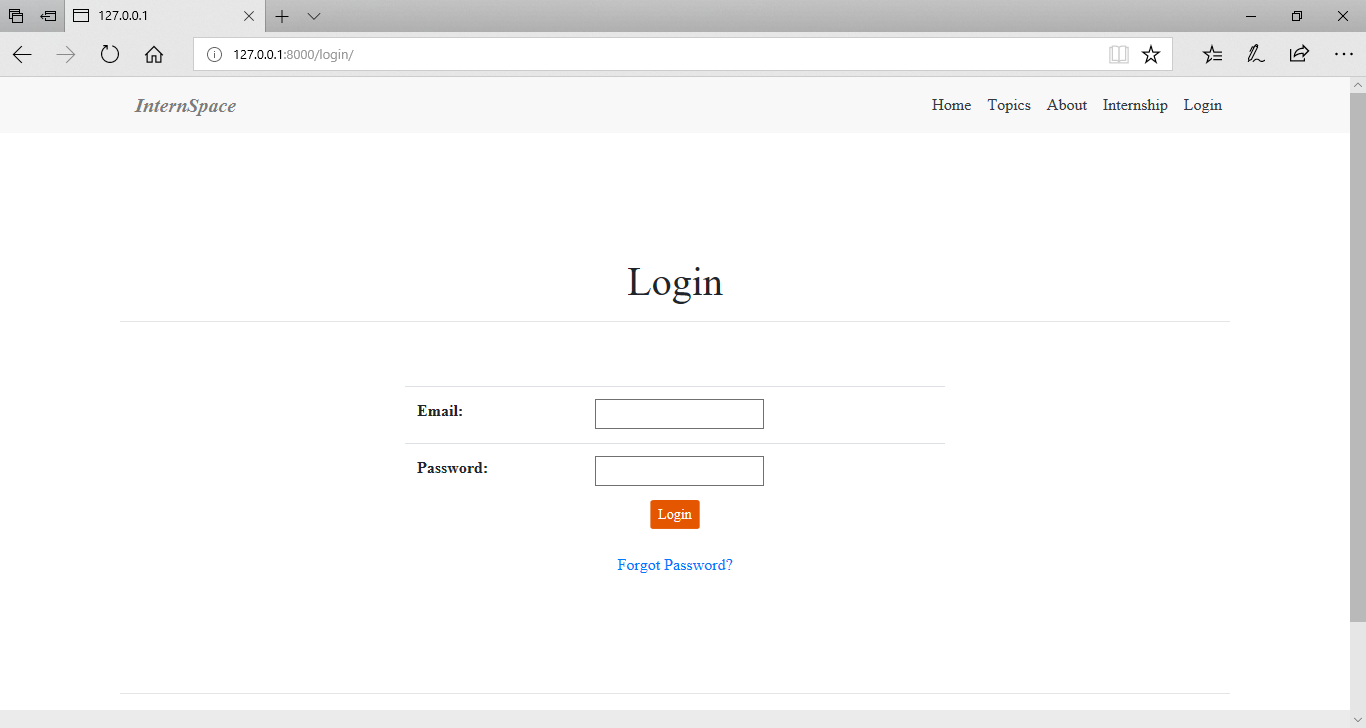
            </div>

        </center>

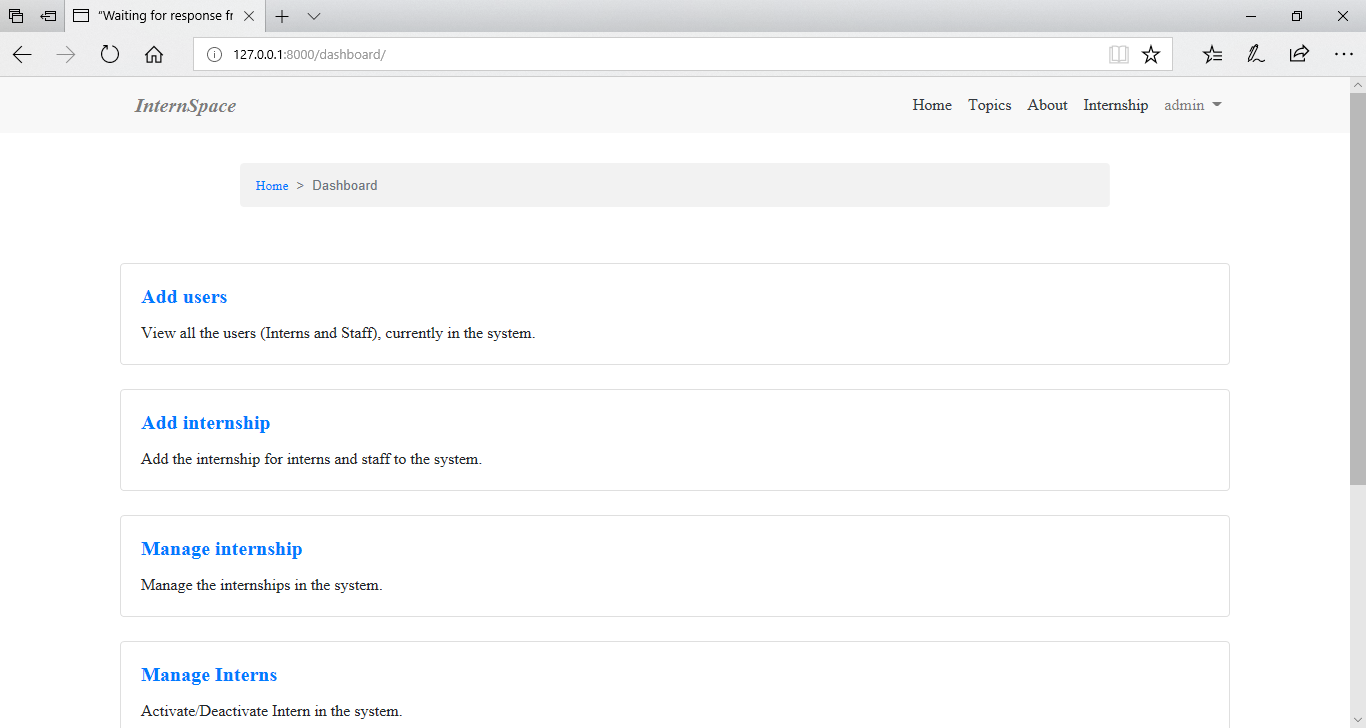
    </div> {% endblock %}

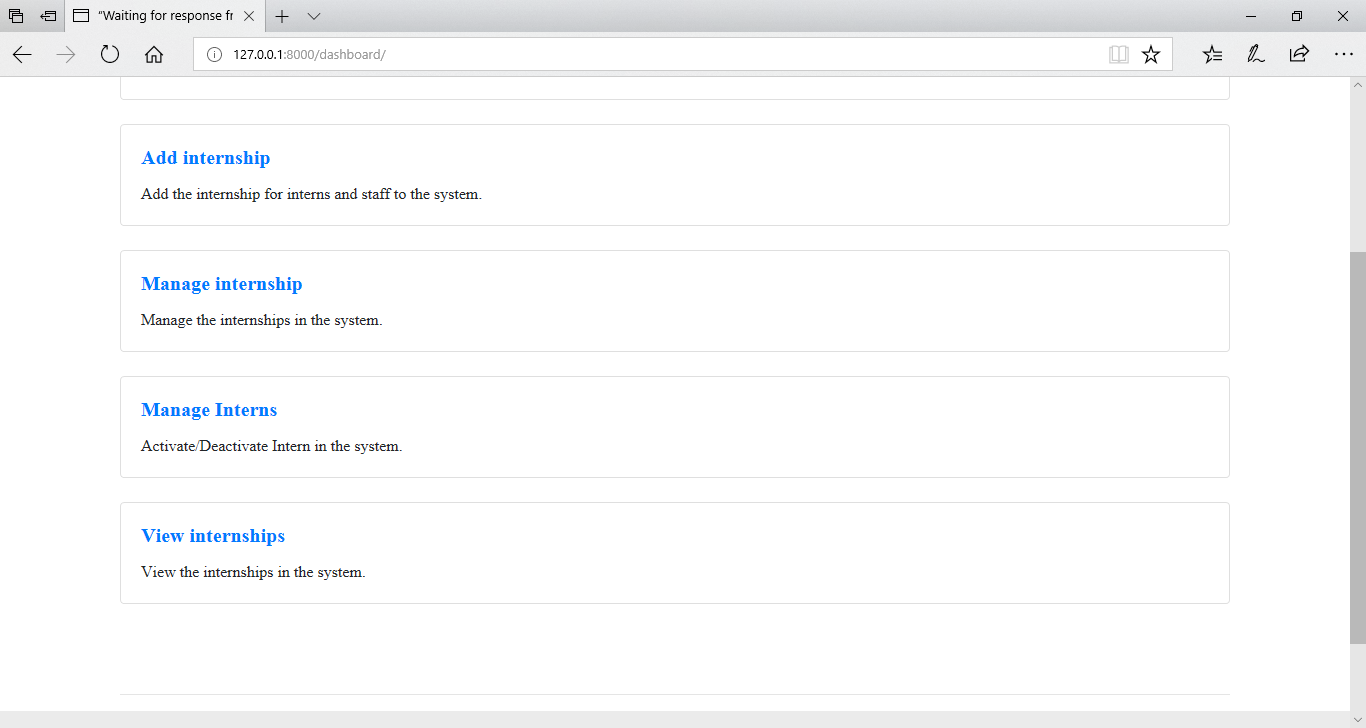
**SCREENSHOTS**

**ADMIN LOGIN**

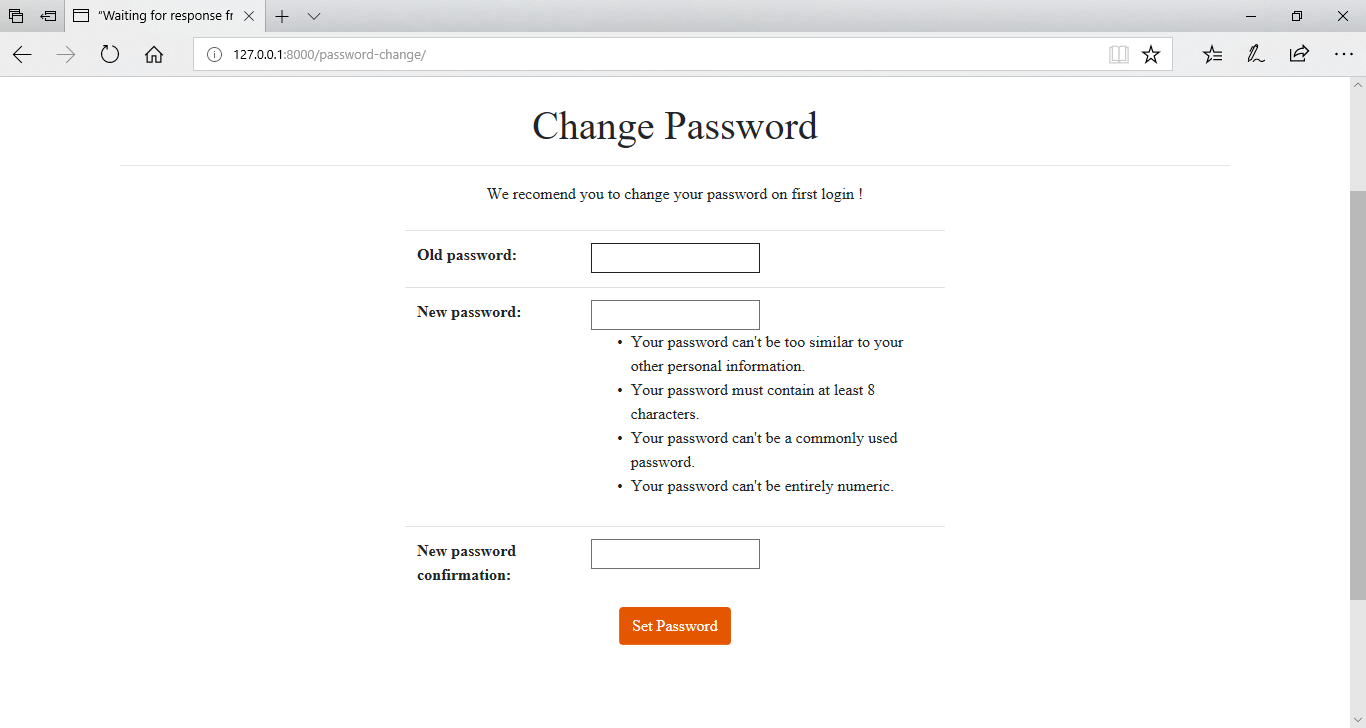
****

**DASHBOARD**

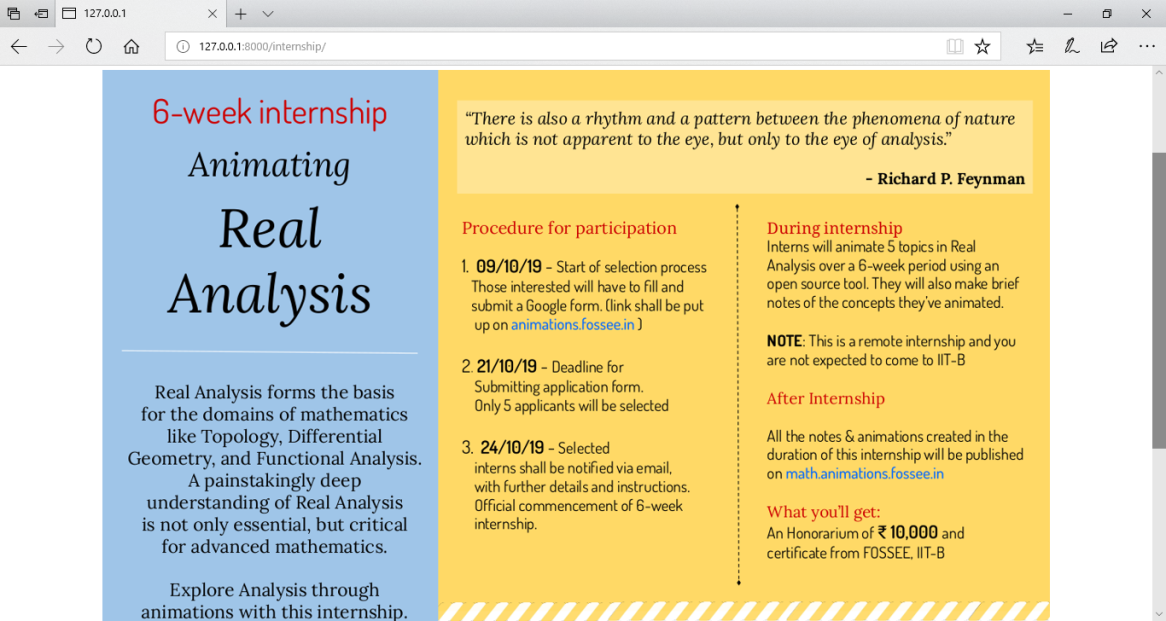
****

****

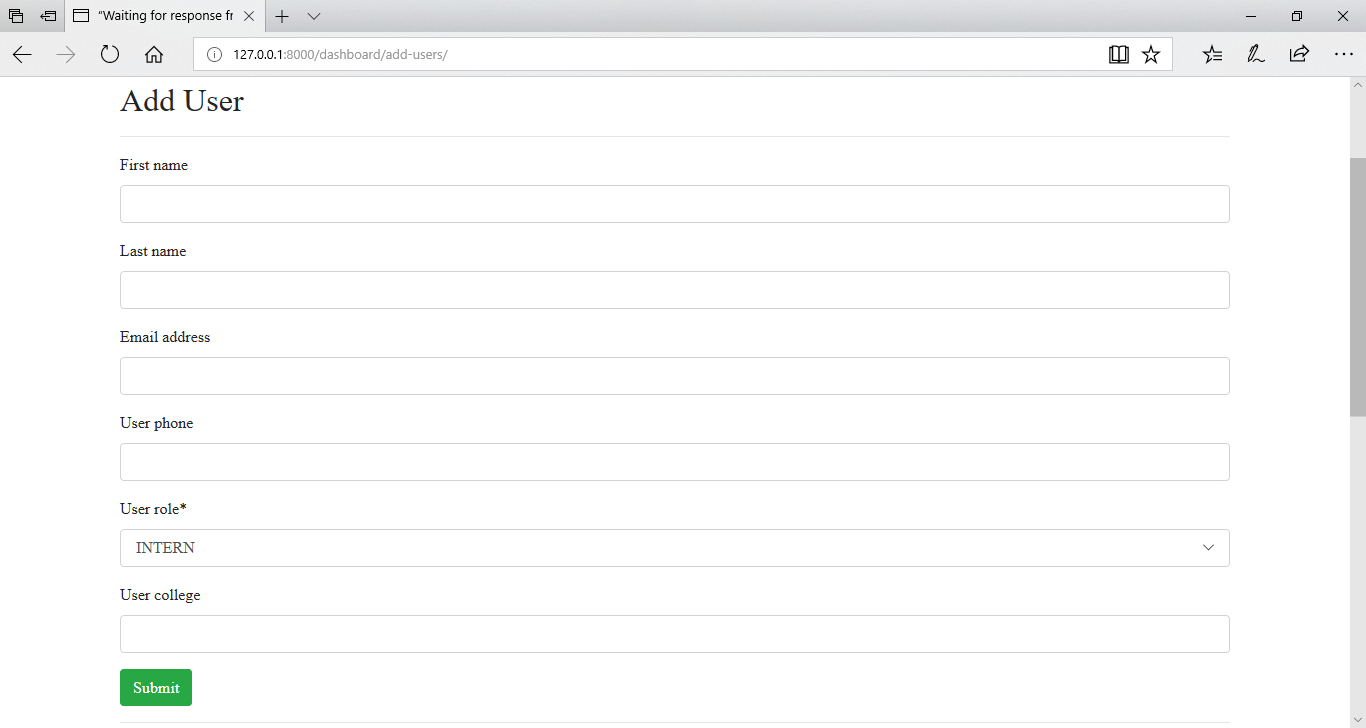
**CHANGE PASSWORD**

****

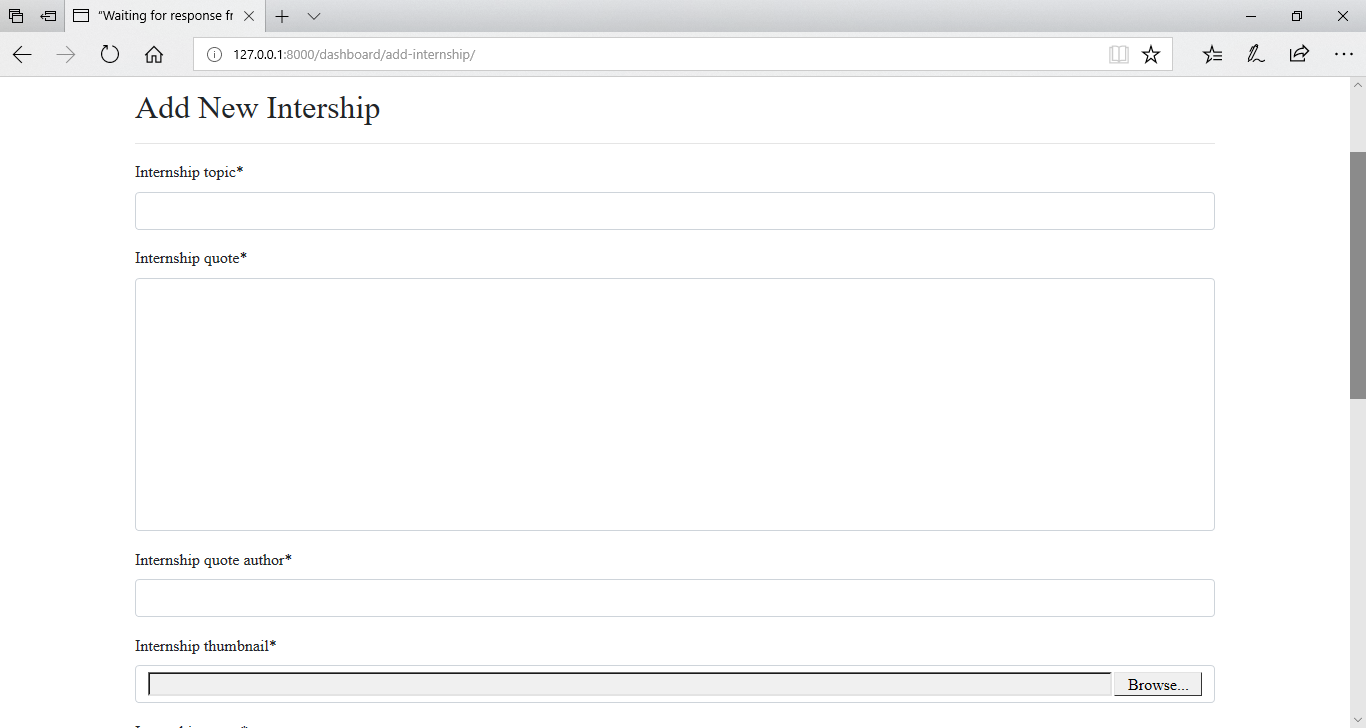
**INTERNSHPS**

****

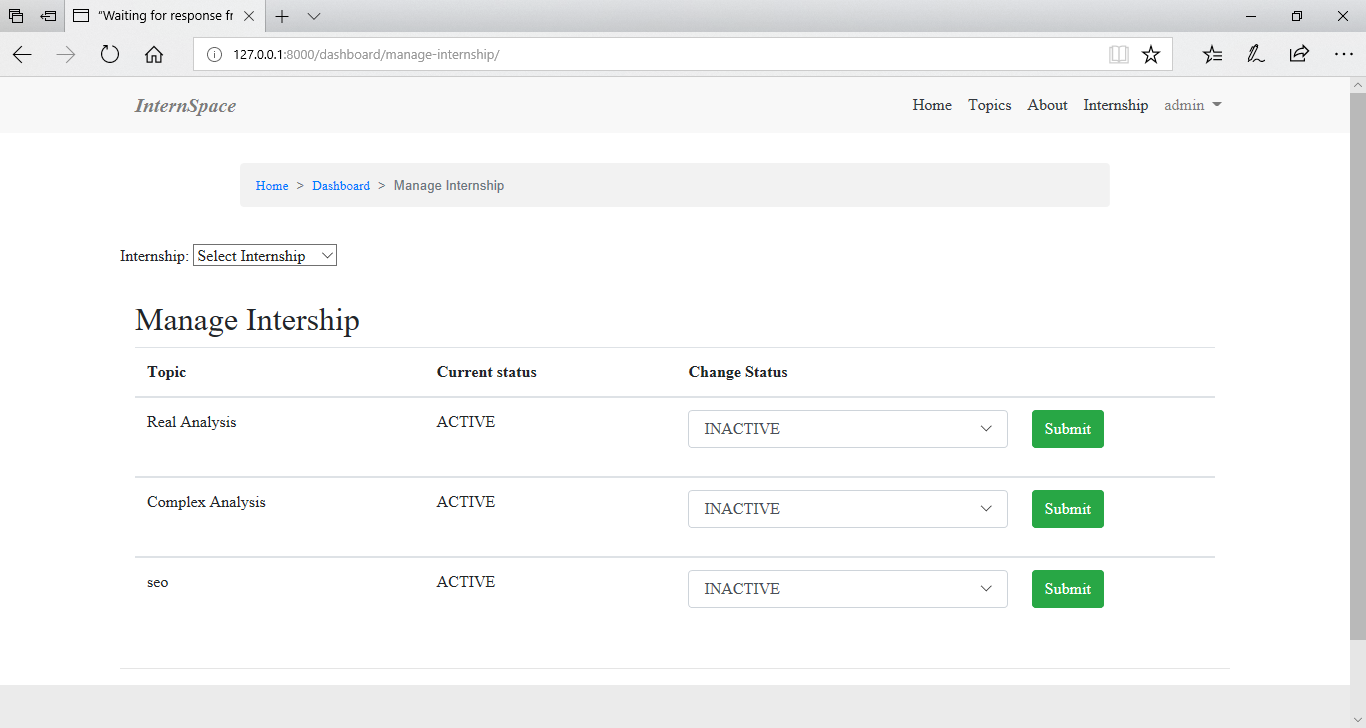
**ADD USER**

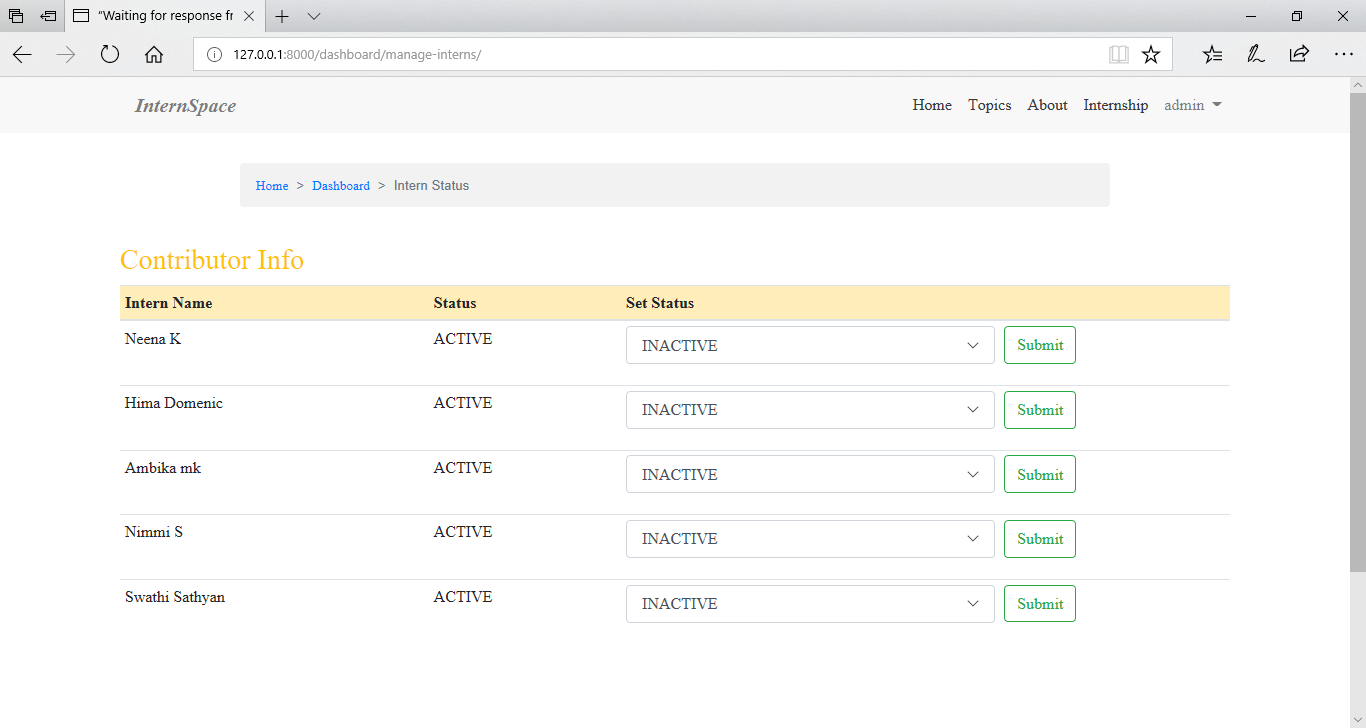
****

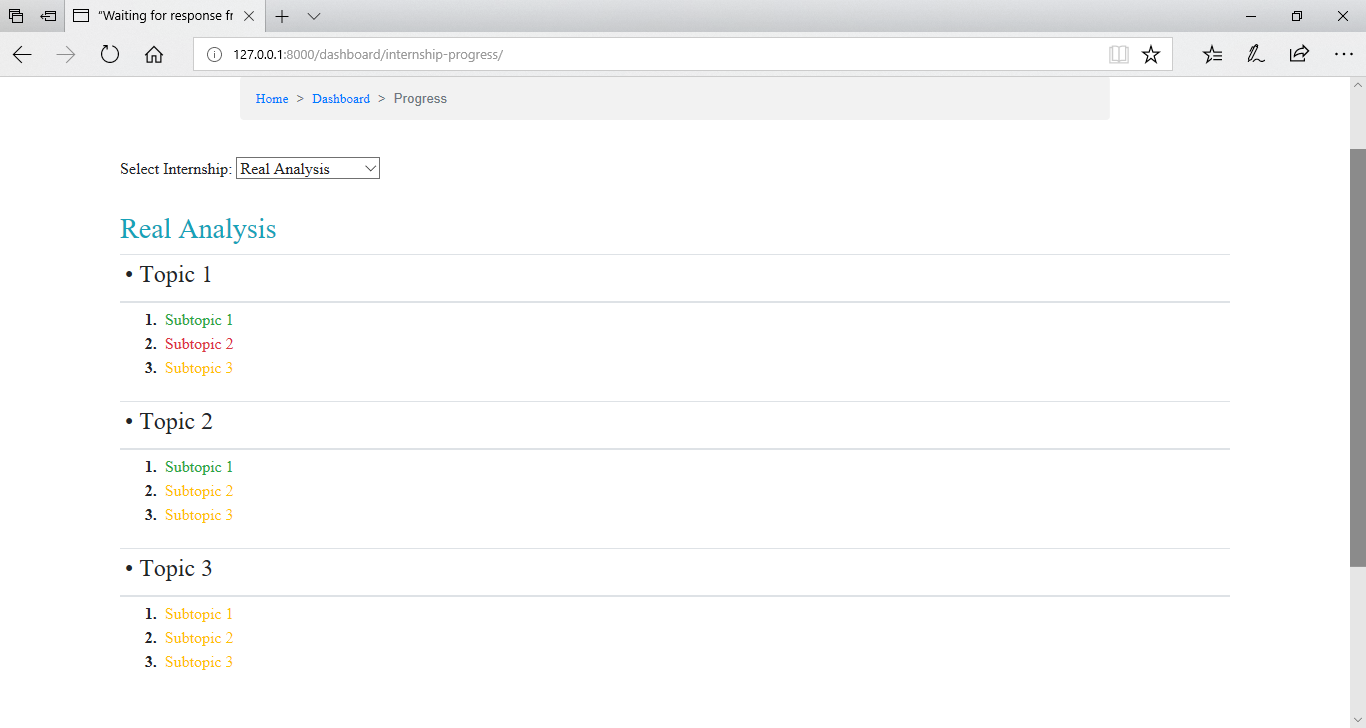
**ADD INTERNSHIPS**

****

**MANAGE INTERNSHPS**

****

****

****