Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering

Department of Computer Engineering

Project Report

On

THINGS TO REMEMBER APP

Submitted by,

Sujeet Jawale BECOA138

Adwait Deshmukh BECOA124

Sangram Deshmukh BECOA123

DEPARTMENT OF COMPUTER ENGINEERING,
PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
Sector No. 26, Pradhikaran, Nigdi,
Pune - 411044

INDEX

Introduction

1. Motivation :

The motivation of this project is to build a web application for tracking status of daily tasks set by the user and maintain its data. This will help users to add, delete tasks in task tracker application and also inform users how many tasks are incomplete and complete.

2. Objectives :

The main objective is to build a web application that can perform following operations on task tracker app:
1] Add tasks
2] Add Description about task
3] Delete tasks from task list
4] Show the count of incomplete tasks

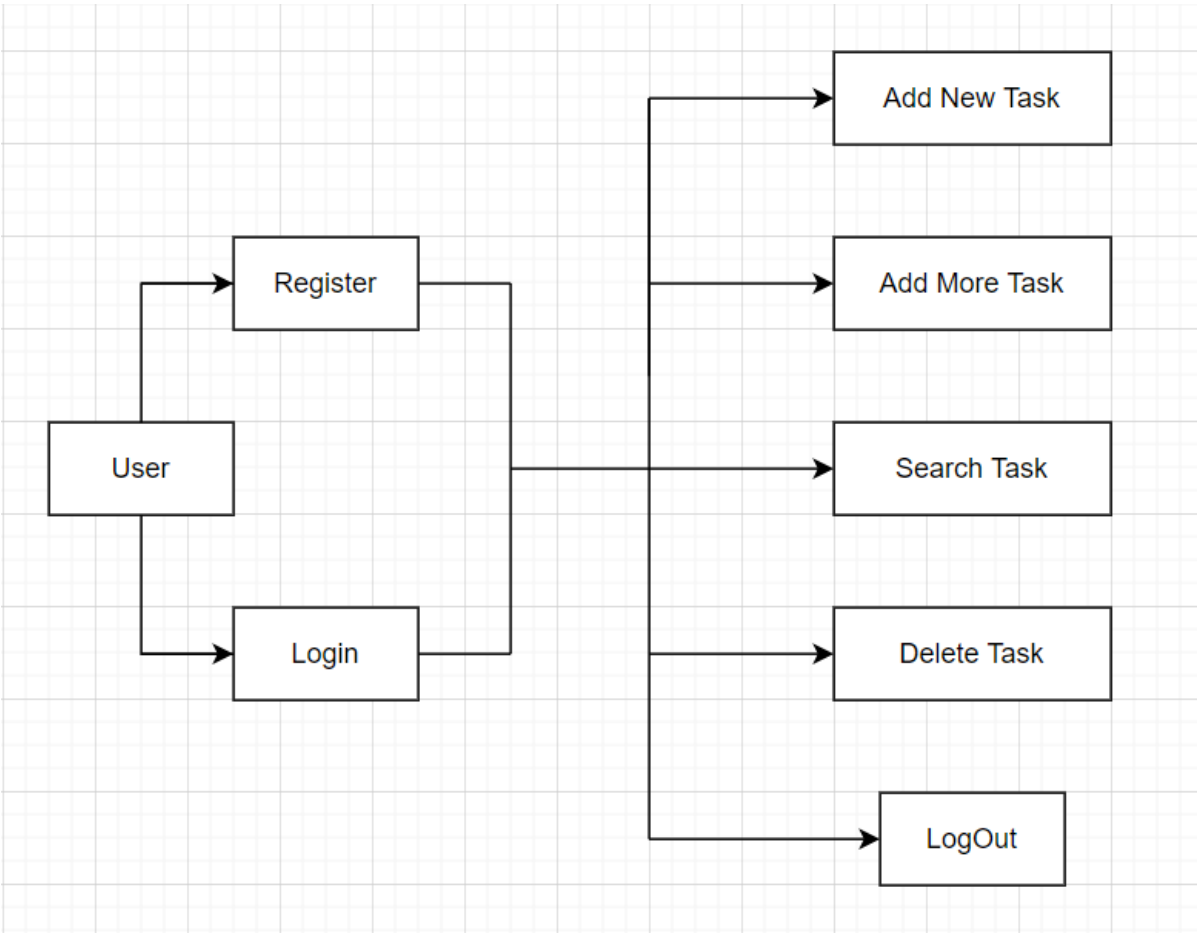
Functional Requirements :

1] User can be register successfully
2] User can be login successfully
3] Task can be added into tasklist and database
4] Description about task can be added successfully
5] Task can be deleted from task and database
6] Users will get to know about count of incomplete task
7] User can be log out successfully

Non-Functional Requirements :

1] Performance should be fast
2] App should be easy to use
3] Database should be normalized
4] Errors should be handled properly
5] Software should pass all the tests during testing

Block Diagram :



Source code :

1 login.html

```
{% extends 'base/main.html' %}
{% block content %}

<div class="header-bar">
   <h1>Login</h1>
</div>

<div class="card-body">
   <form method="POST">
      {% csrf_token %}
      {{form.as_p}}
      <input class="button" type="submit" value="Login">
   </form>
   <p>Don't have an account? <a href="{% url 'register' %}">Register</a></p>
</div>

{% endblock content %}
```

2 main.html

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="utf-8">
   <meta name="description"
      content="This is to do list implemented using Django by Dennis Ivy who is a full stack web
developer.">

   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <title>TaskTracker</title>
   <meta name="description" content="">
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link rel="preconnect" href="https://fonts.gstatic.com">
   <link href="https://fonts.googleapis.com/css2?family=Nunito:wght@200&display=swap"
rel="stylesheet">

   <style>
      body {
         background-color: #FAFAFA;
         font-family: 'Nunito', sans-serif;
         padding-top: 50px;
```

```
}

h1,
h2,
h3,
h4,
h5,
h6,
   {
   font-family: 'Raleway', sans-serif;
}

a,
p {
   color: #4b5156
}

a {
   font-weight: 600;
}

.container {
   max-width: 550px;
   margin: auto;
   background-color: #fff;
   -webkit-box-shadow: 2px 2px 13px -4px rgba(0, 0, 0, 0.21);
   box-shadow: 2px 2px 13px -4px rgba(0, 0, 0, 0.21);
}

input {
   outline: none;
   border: none;
}

.header-bar {
   display: flex;
   justify-content: space-between;
   color: #fff;
   padding: 10px;
   border-radius: 5px 5px 0 0;
   background: linear-gradient(90deg, #EEA390 0%, #EB796F 43%, #EB796F 100%);
}

.header-bar a {
   color: rgb(247, 247, 247);
   text-decoration: none;
}
```

```
.task-wrapper {
    display: flex;
    align-items: center;
    justify-content: space-between;
    background-color: #fff;
    border-top: 1px solid #dfe4ea;
    overflow: hidden;
}

.task-title {
    display: flex;
    padding: 20px;
}

.task-title a {
    text-decoration: none;
    color: #4b5156;
    margin-left: 10px;
}

.task-complete-icon {
    height: 20px;
    width: 20px;
    background-color: rgb(105, 192, 105);
    border-radius: 50%;
}

.task-incomplete-icon {
    height: 20px;
    width: 20px;
    background-color: rgb(218, 218, 218);
    border-radius: 50%;
}

.delete-link {
    text-decoration: none;
    font-weight: 900;
    color: #cf4949;
    font-size: 22px;
    line-height: 0;
    padding: 20px 0px;
}

/*Handle Classes*/

.handle {
    display: inline-block;
    padding: 20px 16px;
```

```
    cursor: grab;
    user-select: none;
}

.handle:after,
.handle:before {
    display: block;
    content: "";
}

.handle:active,
.handle:active:before,
.handle:active:after {
    cursor: grabbing;
}

.dropArea {
    background-color: #f1f2f6;
    color: black;
    border: #ced6e0 1px solid;
}

.selectedTask {
    opacity: 0.6;
}


#add-link {
    color: #EB796F;
    text-decoration: none;
    font-size: 42px;
    text-shadow: 1px 1px #81413b;
}

#search-add-wrapper {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 20px;
}

@media screen and (max-width:320px) {
    #search-add-wrapper {
        padding: 10px;
    }
}

input[type=text],
```

```
        input[type=password],
        textarea {
            border: 1px solid #757575;
            border-radius: 5px;
            padding: 10px;
            width: 90%;
        }

        label {
            padding-top: 10px !important;
            display: block;
        }

        ::placeholder {
            font-weight: 300;
            opacity: 0.5;
        }

        .button {
            border: 1px solid #757575;
            background-color: #FFF;
            color: #EB796F;
            padding: 10px;
            font-size: 14px;
            border-radius: 5px;
            cursor: pointer;
            text-decoration: none;
        }

        .card-body {
            padding: 20px;
        }
    </style>
</head>
<body>
    <div class="container">
        {% block content %} {% endblock content %}
    </div>
</body>
</html>
```

3 register.html

```
{% extends 'base/main.html' %}
{% block content %}

<div class="header-bar">
    <h1>Register</h1>
</div>
```

```
<div class="card-body">
   <form method="POST">
      {% csrf_token %}
      <label>{{form.username.label}}</label>
      {{form.username}}

      <label>{{form.password1.label}}</label>
      {{form.password1}}

      <label>{{form.password2.label}}</label>
      {{form.password2}}
      <input style="margin-top:10px ;" class="button" type="submit" value="Register">
   </form>
   <p>Already have an account? <a href="{% url 'login' %}">Login</a></p>
</div>

{% endblock content %}
```

4 task.html

```
<h1>Task: {{task}}</h1>
```

5 task_confirm_delete.html

```
{% extends 'base/main.html' %}
{% block content %}

<div class="header-bar">
   <a href="{% url 'tasks' %}">&amp;#8592; Go Back</a>
</div>
<div class="card-body">
   <form method="POST">
      {% csrf_token %}
      <p>Are your sure you want to delete this task? <b>"{{task}}"</b> </p>
      <input class="button" type="submit" value="Delete" />
   </form>
</div>

{% endblock content %}
```

6 task_form.html

```
{% extends 'base/main.html' %}
{% block content %}

<div class="header-bar">
   <a href="{% url 'tasks' %}">&amp;#8592; Back</a>
</div>
<div class="card-body">
   <form method="POST" action="">
      {% csrf_token %}
```

```
        {{form.as_p}}
        <input class="button" type="submit" value="Submit">
    </form>
</div>

{% endblock content %}
```

7 task_list.html

```
{% extends 'base/main.html' %} {% block content %}

<script src="https://cdn.jsdelivr.net/npm/sortablejs@latest/Sortable.min.js"></script>
<div class="header-bar">
    <div>
        <h1>Hello {{request.user|title}}</h1>
        <h3 style="margin:0">You have <i>{{count}}</i> incomplete task{{ count|pluralize:"s"
}}</h3>
    </div>
    {% if request.user.is_authenticated %}
    <a id="logout" href="{% url 'logout' %}">Logout</a> {% else %}
    <a href="{% url 'login' %}">Login</a> {% endif %}
</div>
<div id="search-add-wrapper">
    <form method="GET" style="display: flex;">
        <input type='text' name='search-area' placeholder="Search your task"
value="{{search_input}}">
        <input class="button" type="submit" value='Search'>
    </form>
    {% if tasks|length > 0 %}
    <a id="addlink" href="{% url 'task-create' %}">&amp;#x2b;</a>
    {% endif %}
</div>
<form style="display: none;" id="reorderForm" method="post" action="{% url 'task-reorder' %}">
    {% csrf_token %}
    <input type="hidden" id="positionInput" name="position">
</form>


<div id="tasklist" class="task-items-wrapper">
    {% for task in tasks %}
    <div class="task-wrapper" data-position="{{task.pk}}">
        <div class="task-title" >
            {% if task.complete %}
            <div class="task-complete-icon"></div>
            <i><s><a id="" href="{% url 'task-update' task.id %}">{{task}}</a></s></i> {% else %}
            <div class="task-incomplete-icon" ></div>
            <a href="{% url 'task-update' task.id %}">{{task}}</a> {% endif %}
        </div>
        <div class="task-controls">
```

```
            <a id="deletelink" class="delete-link" href="{% url 'task-delete' task.id %}">&#215;</a>
            <span class="handle"> &#10247;</span>
        </div>
    </div>

    {% empty %}
    <div style="text-align: center; padding-bottom: 10px; line-height: 1em;">
        <h3>No new tasks are created.</h3>
        <h3 id="newtask"> <a style="text-decoration: none; color: #e53935;" href="{% url 'task-create'
%}">New task</a> ! </h3>
    </div>
    {% endfor %}
</div>

<script>
    var taskList = document.getElementById("tasklist");
    var reorderForm = document.getElementById("reorderForm");
    var positionInput = document.getElementById("positionInput");

    let sortable = Sortable.create(taskList, {
        handle: '.handle',
        ghostClass: 'dropArea',
        chosenClass: 'selectedTask',

    });
    function reordering() {
        const rows = document.getElementsByClassName("task-wrapper");
        let pos = [];
        for (let row of rows) {
            pos.push(row.dataset.position);
        }
        console.log(pos.join(","))
        positionInput.value = pos.join(',');
        reorderForm.submit();
    }
    document.ondrop = reordering
</script>

{% endblock content %}
```

8. admin.py

```
from django.contrib import admin
from .models import Task

admin.site.register(Task)
```

9.forms.py

```
from django import forms
```

```
class PositionForm(forms.Form):
    position = forms.CharField()
```

10. urls.py

```
from django.urls import path
from .views import TaskList, TaskDetail, TaskCreate, TaskUpdate, DeleteView, CustomLoginView,
RegisterPage, TaskReorder
from django.contrib.auth.views import LogoutView

urlpatterns = [
    path('login/', CustomLoginView.as_view(), name='login'),
    path('logout/', LogoutView.as_view(next_page='login'), name='logout'),
    path('register/', RegisterPage.as_view(), name='register'),

    path('', TaskList.as_view(), name='tasks'),
    path('task/<int:pk>/', TaskDetail.as_view(), name='task'),
    path('task-create/', TaskCreate.as_view(), name='task-create'),
    path('task-update/<int:pk>/', TaskUpdate.as_view(), name='task-update'),
    path('task-delete/<int:pk>/', DeleteView.as_view(), name='task-delete'),
    path('task-reorder/', TaskReorder.as_view(), name='task-reorder'),
]
```

11. views.py

```
from django.shortcuts import render, redirect
from django.views.generic.list import ListView
from django.views.generic.detail import DetailView
from django.views.generic.edit import CreateView, UpdateView, DeleteView, FormView
from django.urls import reverse_lazy

from django.contrib.auth.views import LoginView
from django.contrib.auth.mixins import LoginRequiredMixin
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import login

# Imports for Reordering Feature
from django.views import View
from django.shortcuts import redirect
from django.db import transaction

from .models import Task
from .forms import PositionForm


class CustomLoginView(LoginView):
    template_name = 'base/login.html'
    fields = '__all__'
    redirect_authenticated_user = True
```

```
    def get_success_url(self):
        return reverse_lazy('tasks')


class RegisterPage(FormView):
    template_name = 'base/register.html'
    form_class = UserCreationForm
    redirect_authenticated_user = True
    success_url = reverse_lazy('tasks')

    def form_valid(self, form):
        user = form.save()
        if user is not None:
            login(self.request, user)
        return super(RegisterPage, self).form_valid(form)

    def get(self, *args, **kwargs):
        if self.request.user.is_authenticated:
            return redirect('tasks')
        return super(RegisterPage, self).get(*args, **kwargs)


class TaskList(LoginRequiredMixin, ListView):
    model = Task
    context_object_name = 'tasks'

    def get_context_data(self, **kwargs):
        context =  super().get_context_data(**kwargs)
        context['tasks'] = context['tasks'].filter(user=self.request.user)
        context['count'] = context['tasks'].filter(complete=False).count()

        search_input = self.request.GET.get('search-area') or ''
        if search_input:
            context['tasks'] = context['tasks'].filter(
                title_contains=search_input)

        context['search_input'] = search_input

        return context


class TaskDetail(LoginRequiredMixin, DetailView):
    model = Task
    context_object_name = 'task'
    template_name = 'base/task.html'
```

```
class TaskCreate(LoginRequiredMixin, CreateView):
    model = Task
    fields = ['title', 'description', 'complete']
    success_url = reverse_lazy('tasks')

    def form_valid(self, form):
        form.instance.user = self.request.user
        return super(TaskCreate, self).form_valid(form)


class TaskUpdate(LoginRequiredMixin, UpdateView):
    model = Task
    fields = ['title', 'description', 'complete']
    success_url = reverse_lazy('tasks')


class DeleteView(LoginRequiredMixin, DeleteView):
    model = Task
    context_object_name = 'task'
    success_url = reverse_lazy('tasks')
    def get_queryset(self):
        owner = self.request.user
        return self.model.objects.filter(user=owner)

class TaskReorder(View):
    def post(self, request):
        form = PositionForm(request.POST)

        if form.is_valid():
            positionList = form.cleaned_data["position"].split(',')

            with transaction.atomic():
                self.request.user.set_task_order(positionList)

        return redirect(reverse_lazy('tasks'))
```

12. settings.py

```
import os

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

SECRET_KEY = 'd)=^c7!0-oqjmqve%(bt+p#sq6x*ipz2keh741j*-@f@_)f!1t'
DEBUG = True

ALLOWED_HOSTS = []
```

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'base.apps.BaseConfig',
]

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 = 'todo_list.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        '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',
            ],
        },
    },
]

WSGI_APPLICATION = 'todo_list.wsgi.application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

```
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',
    },
]

LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True
LOGIN_URL = 'login'
STATIC_URL = '/static/'
```

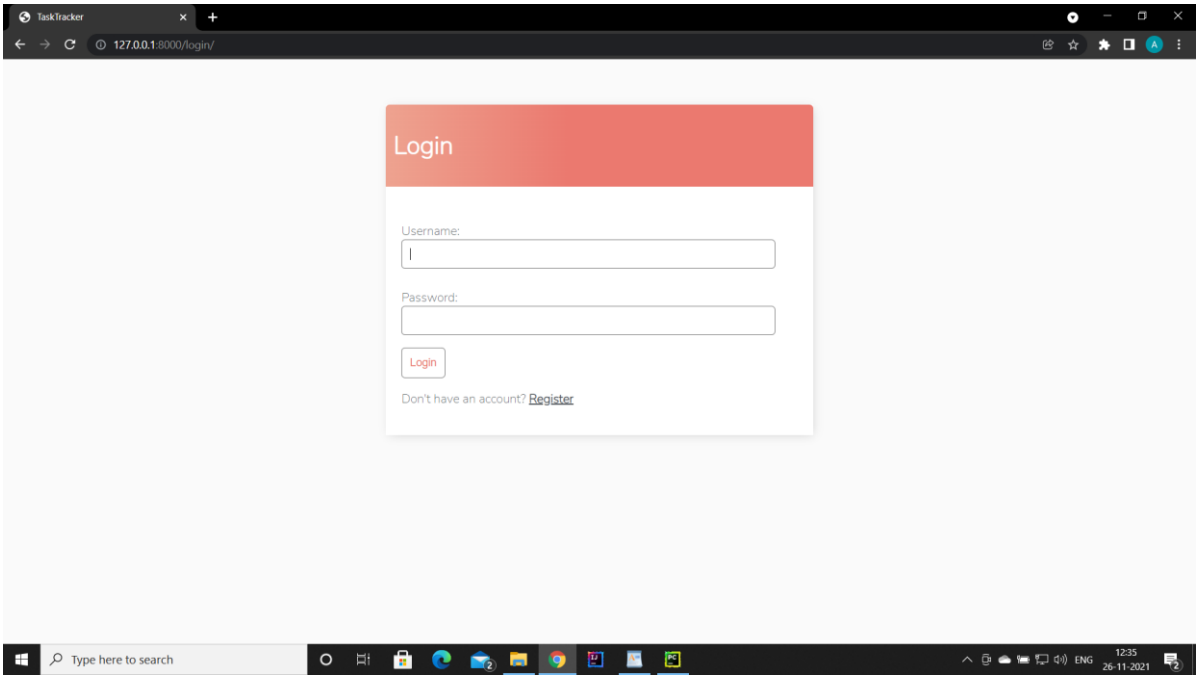
13 urls.py

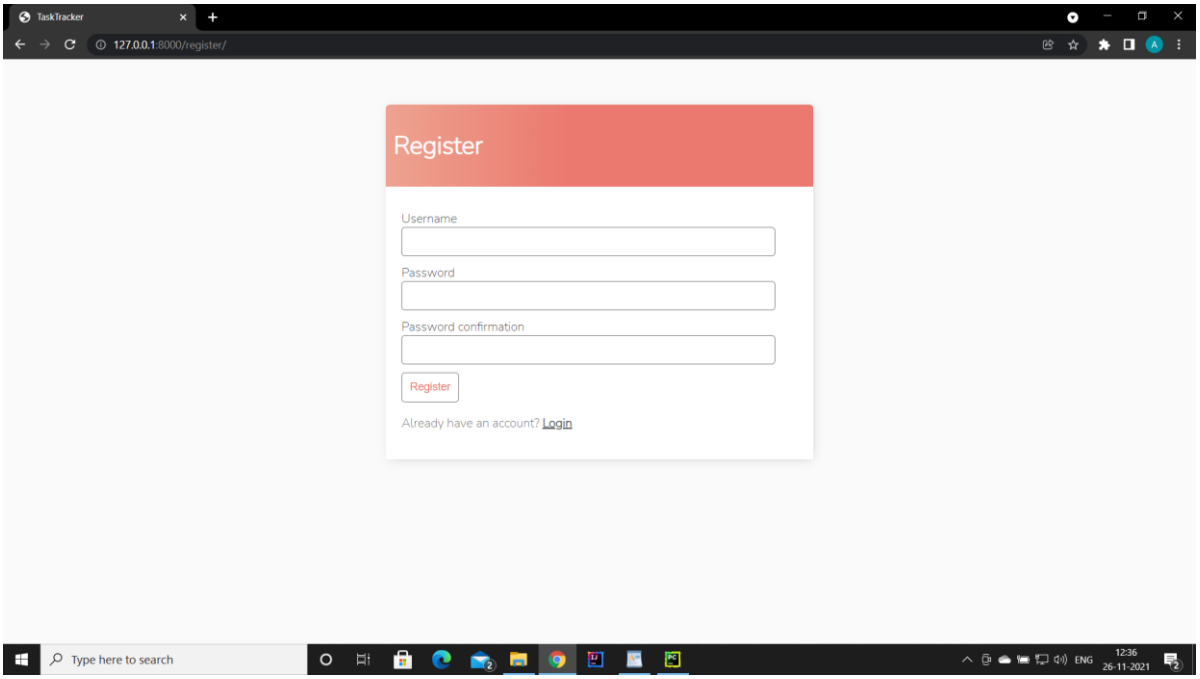
```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('base.urls')),
]
```

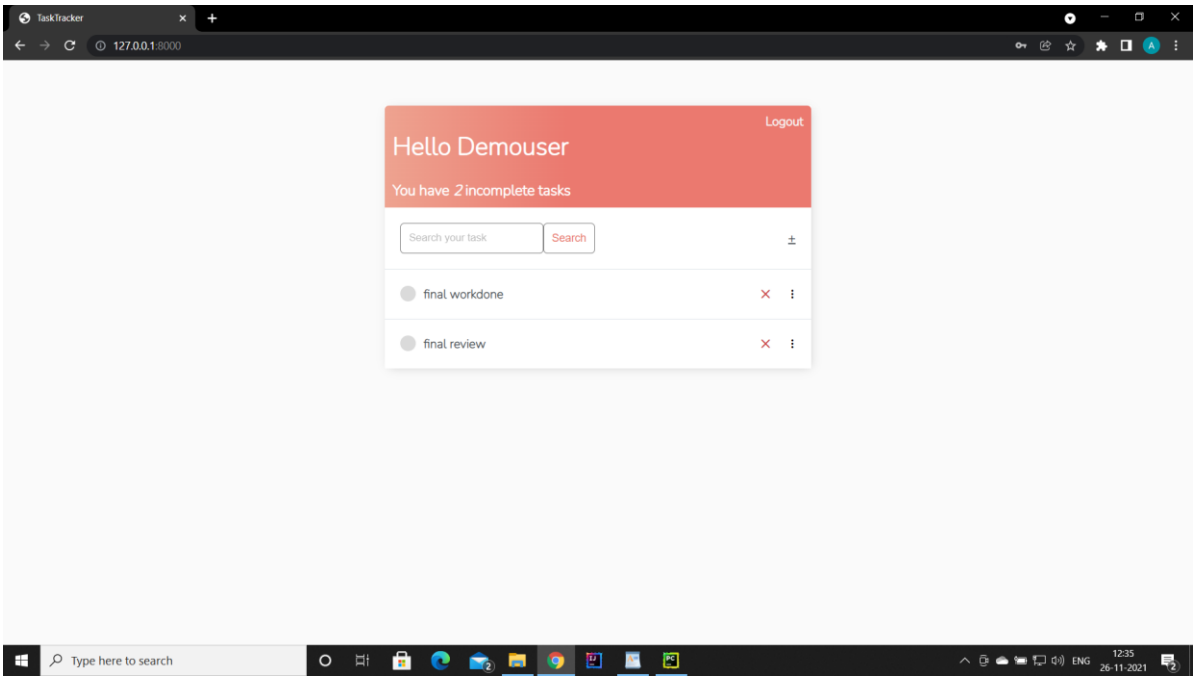
14.wsgi.py

```
import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'todo_list.settings')

application = get_wsgi_application()
```

Output Screenshots :


TaskTracker
127.0.0.1:8000/login/
Login
Username:
Password:
Login
Don't have an account? Register
Type here to search
ENG
12:35
26-11-2021


TaskTracker
127.0.0.1:8000/register/
Register
Username
Password
Password confirmation
Register
Already have an account? Login
Type here to search
ENG
12:36
26-11-2021


TaskTracker
127.0.0.1:8000
Logout
Hello Demouser
You have 2 incomplete tasks
Search your task
Search
final workdone
final review
Type here to search
ENG
12:35
26-11-2021

Test Plan Details

Intended Audience :

• User

Test Scope :

• Functionality Testing,
• User Interface Testing

Assumptions of Testing :

• User is able to understand English
• User is comfortable with windows
• User is able to use web application
• Application is platform independent

Risk Analysis :

• User not able to understand English will not be able to use this application.
• User will not able to use application without credentials i.e username and password
• User may enter invalid details and may cause further issues.

Test Scenarios :

• TS_001 – User is able to register
• Ts_002– User is able to login
• TS_003 – User is able to add task in task list
• TS_004 – User is able to delete task from task list
• TS_005 – User is able to log out

Test Cases

Test Cases

Test scenarios	Test cases	Use case description	Expected Result	Actual Result	Pass/ Fail
TS_001	1	User enters username to be use	User should be able to create set username	Username is save in database	Pass
	2	User enters password to be use	User should be able to create password	password is save in database Associates with username	Pass
	3	User re-enter the password to be confirm	User should be able to confirm and password	Password is verified with previously entered password to confirm and save in database	Pass
	4	User enters username to be use	User should be to create username	Username is save in database	Pass
TS_002	5	User enters valid username and password	User should be able to login the application	User is logged in successfully	Pass
TS_003	6	User enters the task title, description and click on submit	Task should be added to task list	Task added to task list successfully	Pass
	7	User can add multiple tasks to taklist by clicking on plus sign icon to	User should be able to add multiple task in task list	More tasks added in task list successfully	Pass
TS_004	8	User can delete task from task list by selecting task and give confirmation for deletion of task	User should able to delete any task from task list	Selected task deleted successfully	Pass
TS_005	9	User can be able to log out by clicking on log out option and exit from application	User should be logged out and able to see home page again	User is logged out successfully and able to see home page again	Pass

Selenium WebDriver Test Cases :

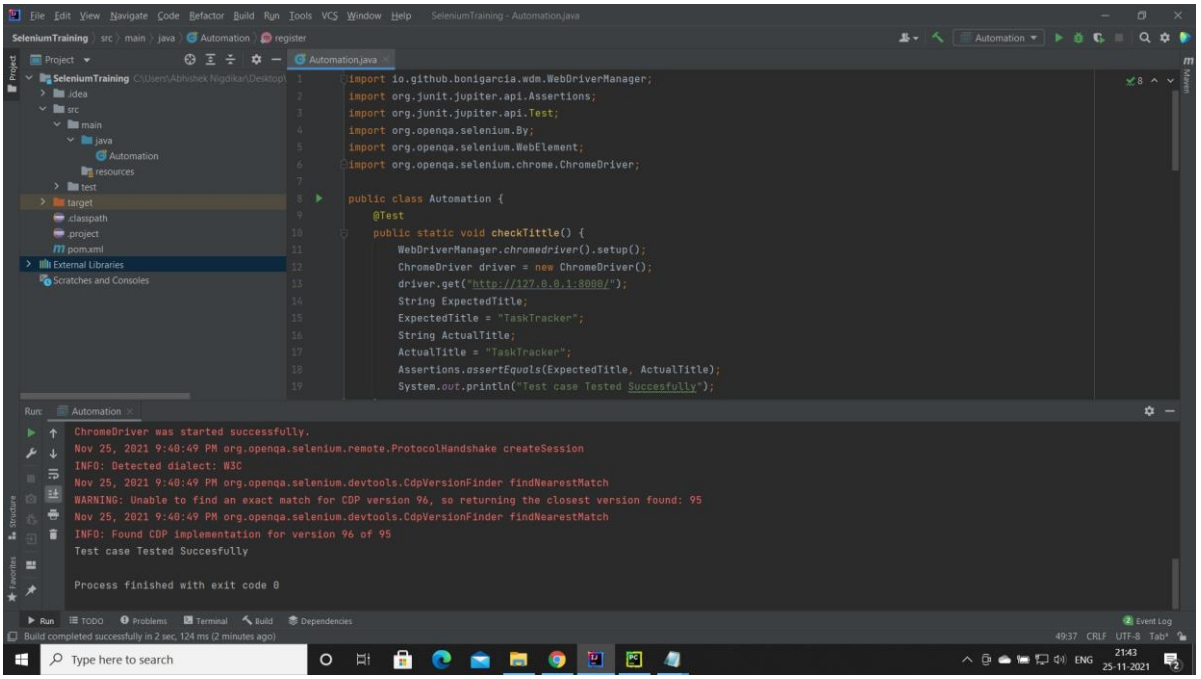
```
import io.github.bonigarcia.wdm.WebDriverManager;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Automation {
	@Test
	public static void checkTittle() {
		WebDriverManager.chromedriver().setup();
		ChromeDriver driver = new ChromeDriver();
		driver.get("http://127.0.0.1:8000/");
		String ExpectedTitle;
		ExpectedTitle = "TaskTracker";
		String ActualTitle;
		ActualTitle = "TaskTracker";
		Assertions.assertEquals(ExpectedTitle, ActualTitle);
		System.out.println("Test case Tested Succesfully");
	}
	@Test
	public static void login() {
		WebDriverManager.chromedriver().setup();
		ChromeDriver driver = new ChromeDriver();
		driver.get("http://127.0.0.1:8000/login/?next=/");
		driver.findElement(By.id("id_username"));
		WebElement username = driver.findElement(By.id("id_username"));
		driver.findElement(By.id("id_password"));
		WebElement password = driver.findElement(By.id("id_password"));
		username.sendKeys("demouser");
		password.sendKeys("demohuman");
		String ExpectedUserName = "demouser";
		String ActualUserName = "demouser";
		Assertions.assertEquals(ExpectedUserName, ActualUserName);
		String ExpectedPassword = "demohuman";
		String ActualPassword = "demohuman";
		Assertions.assertEquals(ExpectedPassword, ActualPassword);
		WebElement login = driver.findElement(By.className("button"));
```

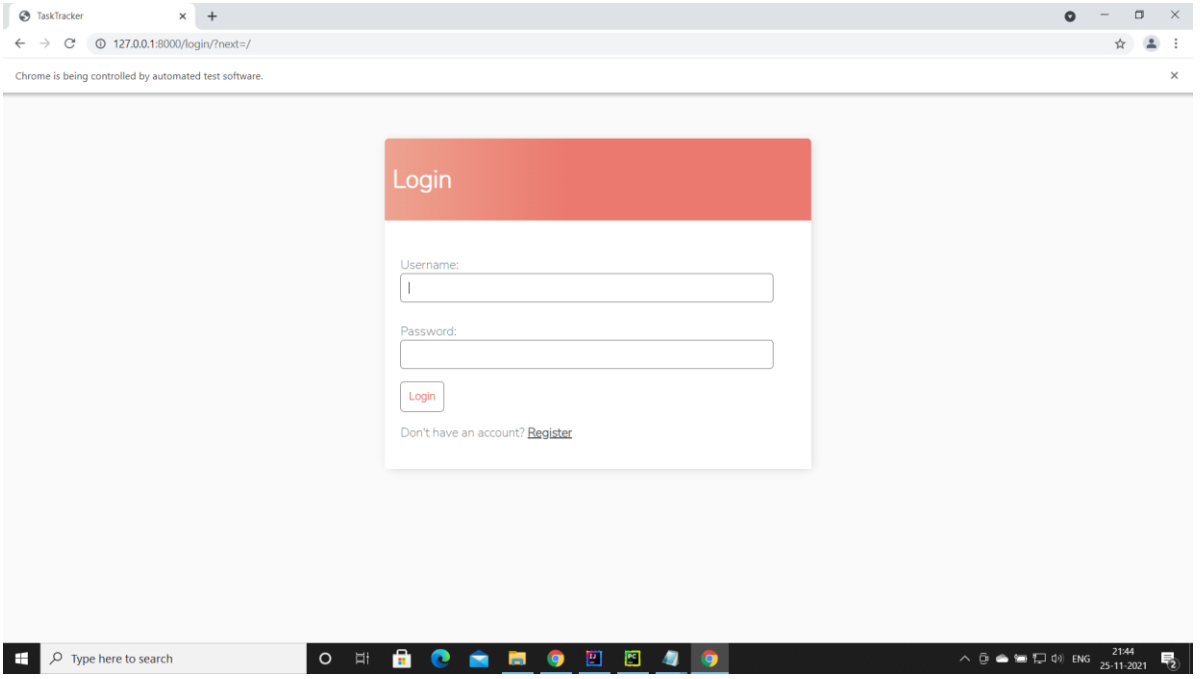
```
		login.click();
		System.out.println("Test case Tested Succesfully");
}
@Test
public static void register() {
		WebDriverManager.chromedriver().setup();
		ChromeDriver driver = new ChromeDriver();
		driver.get("http://127.0.0.1:8000/register/");
		driver.findElement(By.id("id_username"));
		WebElement username = driver.findElement(By.id("id_username"));
		driver.findElement(By.id("id_password1"));
		WebElement password1 = driver.findElement(By.id("id_password1"));
		driver.findElement(By.id("id_password2"));
		WebElement password2 = driver.findElement(By.id("id_password2"));
		username.sendKeys("userone");
		password1.sendKeys("uonepswd");
		password2.sendKeys("uonepswd");
		String ExpectedPassword1 = "uonepswd";
		String ActualPassword1 = "uonepswd";
		Assertions.assertEquals(ExpectedPassword1, ActualPassword1);
		String ExpectedPassword2 = "uonepswd";
		String ActualPassword2 = "uonepswd";
		Assertions.assertEquals(ExpectedPassword2, ActualPassword2);
		WebElement register = driver.findElement(By.className("button"));
		register.click();
		System.out.println("Test case Tested Succesfully");
}
@Test
public static void addTask() {
		String taskone = "myfirstproject";
		String mandatory = "create a website";

		WebDriverManager.chromedriver().setup();
		ChromeDriver driver = new ChromeDriver();
		driver.get("http://127.0.0.1:8000/login/?next=/");
		driver.findElement(By.id("id_username"));
		WebElement username = driver.findElement(By.id("id_username"));
		driver.findElement(By.id("id_password"));
		WebElement password = driver.findElement(By.id("id_password"));
		username.sendKeys("userone");
		password.sendKeys("uonepswd");
		WebElement login = driver.findElement(By.className("button"));
		login.click();
		WebElement newtask = driver.findElement(By.id("newtask"));
```

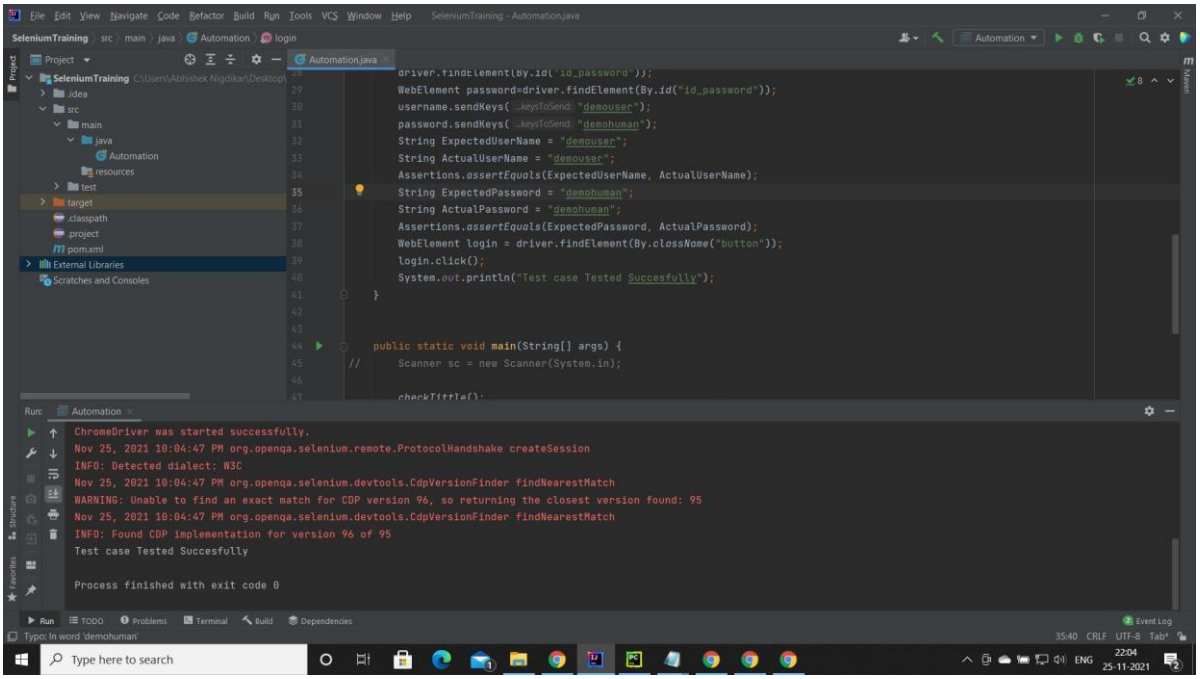
```
        newtask.click();
        driver.findElement(By.id("id_title")).sendKeys(taskone);
        driver.findElement(By.id("id_description")).sendKeys(mandatory);
        WebElement submit = driver.findElement(By.className("button"));
        submit.click();
        System.out.println("Test case Tested Succesfully");
}
@Test
public static void addMore() {
        String tasktwo = "mysecondproject";
        String discription = "Selenium is good";

        WebDriverManager.chromedriver().setup();
        ChromeDriver driver = new ChromeDriver();
        driver.get("http://127.0.0.1:8000/");
        driver.findElement(By.id("id_username"));
        WebElement username = driver.findElement(By.id("id_username"));
        driver.findElement(By.id("id_password"));
        WebElement password = driver.findElement(By.id("id_password"));
        username.sendKeys("userone");
        password.sendKeys("uonepswd");
        WebElement login = driver.findElement(By.className("button"));
        login.click();
        WebElement addlink = driver.findElement(By.id("addlink"));
        addlink.click();
        driver.findElement(By.id("id_title")).sendKeys(tasktwo);
        driver.findElement(By.id("id_description")).sendKeys(discription);
        WebElement submit = driver.findElement(By.className("button"));
        submit.click();
        System.out.println("Test case Tested Succesfully");
}
@Test
public static void delete() {
        WebDriverManager.chromedriver().setup();
        ChromeDriver driver = new ChromeDriver();
        driver.get("http://127.0.0.1:8000/");
        driver.findElement(By.id("id_username"));
        WebElement username = driver.findElement(By.id("id_username"));
        driver.findElement(By.id("id_password"));
        WebElement password = driver.findElement(By.id("id_password"));
        username.sendKeys("userone");
        password.sendKeys("uonepswd");
        WebElement login = driver.findElement(By.className("button"));
        login.click();
```

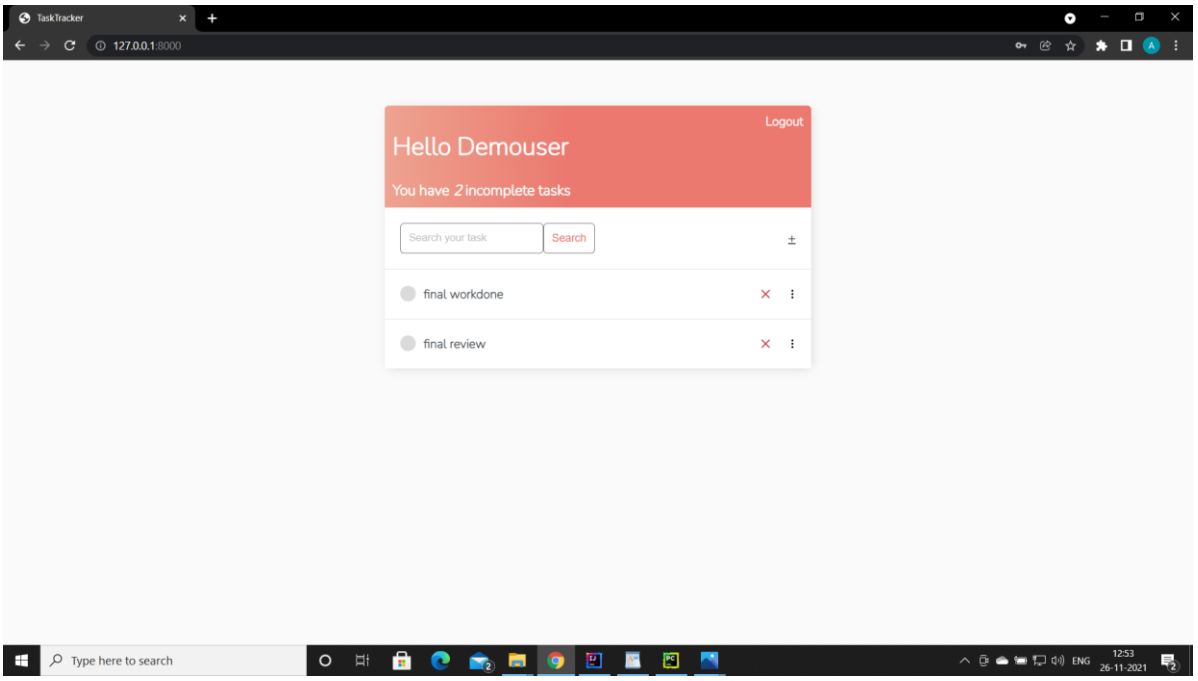
```
		WebElement deletelink = driver.findElement(By.id("deletelink"));
		deletelink.click();
		WebElement  delete = driver.findElement(By.className("button"));
		delete.click();
		System.out.println("Test case Tested Succesfully");
	}
	@Test
	public static void logout() {
		WebDriverManager.chromedriver().setup();
		ChromeDriver driver = new ChromeDriver();
		driver.get("http://127.0.0.1:8000/");
		driver.findElement(By.id("id_username"));
		WebElement username = driver.findElement(By.id("id_username"));
		driver.findElement(By.id("id_password"));
		WebElement password = driver.findElement(By.id("id_password"));
		username.sendKeys("userone");
		password.sendKeys("uonepswd");
		WebElement login = driver.findElement(By.className("button"));
		login.click();
		WebElement logout = driver.findElement(By.id("logout"));
		logout.click();
		System.out.println("Test case Tested Succesfully");
	}
	public static void main(String[] args) {
//		Scanner sc = new Scanner(System.in);

		checkTittle();
		login();
		register();
		addTask();
		addMore();
		delete();
		logout();

	}
}
```

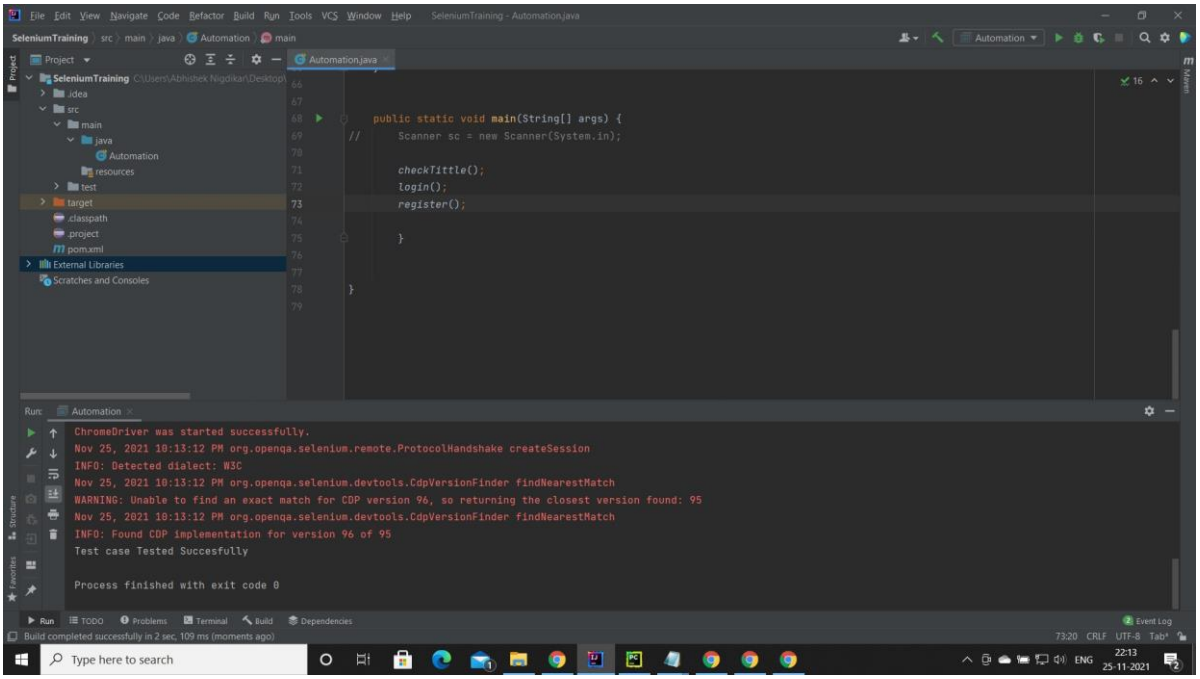
Output Screenshots :

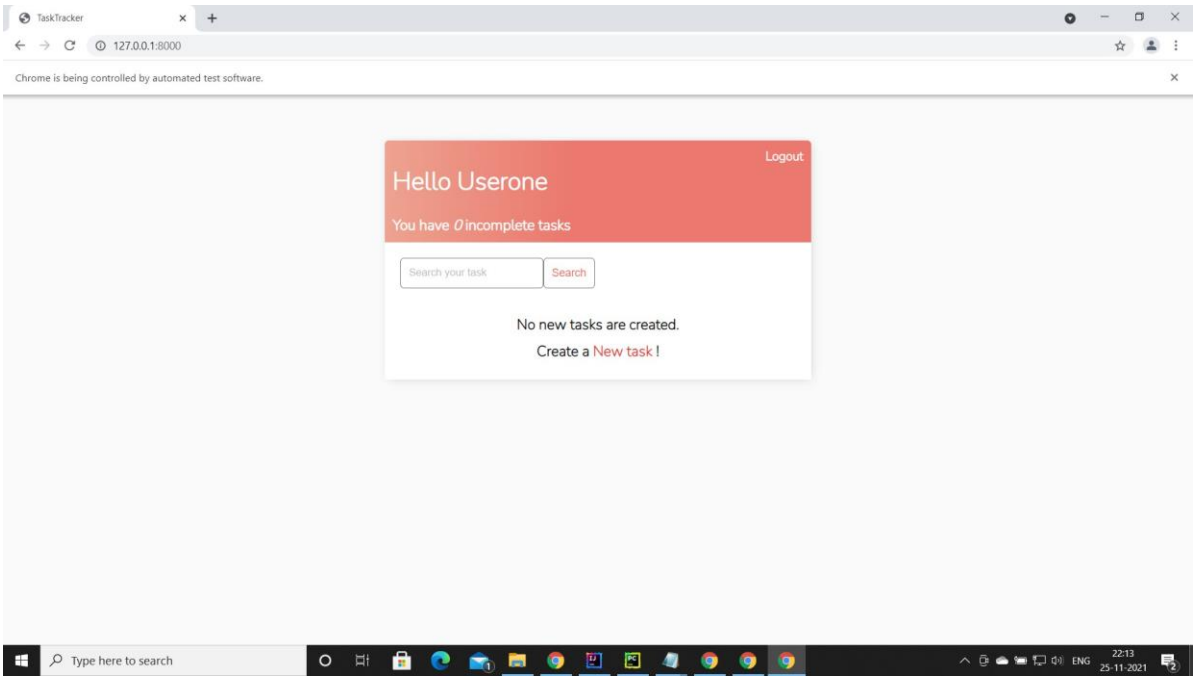

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
SeleniumTraining - Automation.java
SeleniumTraining
Automation
Project
Automation.java
import io.github.bonigarcia.wdm.WebDriverManager;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class Automation {
@Test
public static void checkTittle() {
WebDriverManager.chromedriver().setup();
ChromeDriver driver = new ChromeDriver();
driver.get("http://127.0.0.1:8000/");
String ExpectedTitle;
ExpectedTitle = "TaskTracker";
String ActualTitle;
ActualTitle = "TaskTracker";
Assertions.assertEquals(ExpectedTitle, ActualTitle);
System.out.println("Test case Tested Succesfully");
Run: Automation
ChromeDriver was started successfully.
Nov 25, 2021 9:40:49 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Nov 25, 2021 9:40:49 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 96, so returning the closest version found: 95
Nov 25, 2021 9:40:49 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 96 of 95
Test case Tested Succesfully
Process finished with exit code 0
Build completed successfully in 2 sec, 124 ms (2 minutes ago)
Type here to search
ENG
21:43
25-11-2021

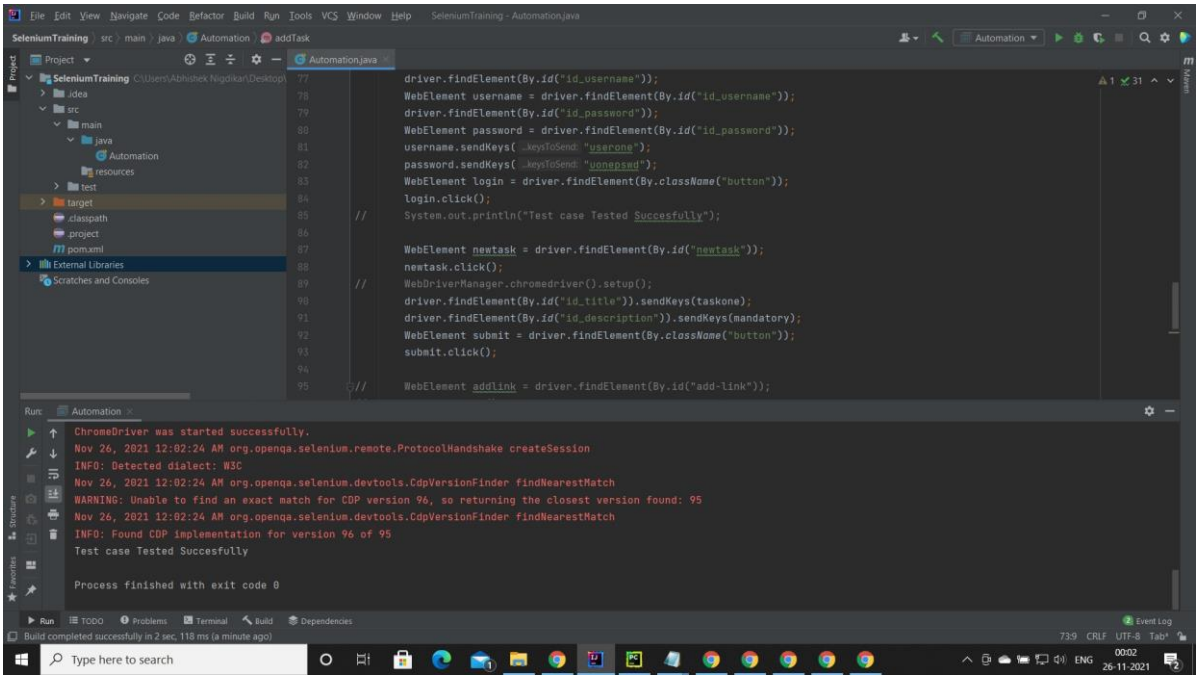

TaskTracker
127.0.0.1:8000/login/?next=/
Chrome is being controlled by automated test software.
Login
Username:
Password:
Login
Don't have an account? Register
Type here to search
ENG
21:44
25-11-2021

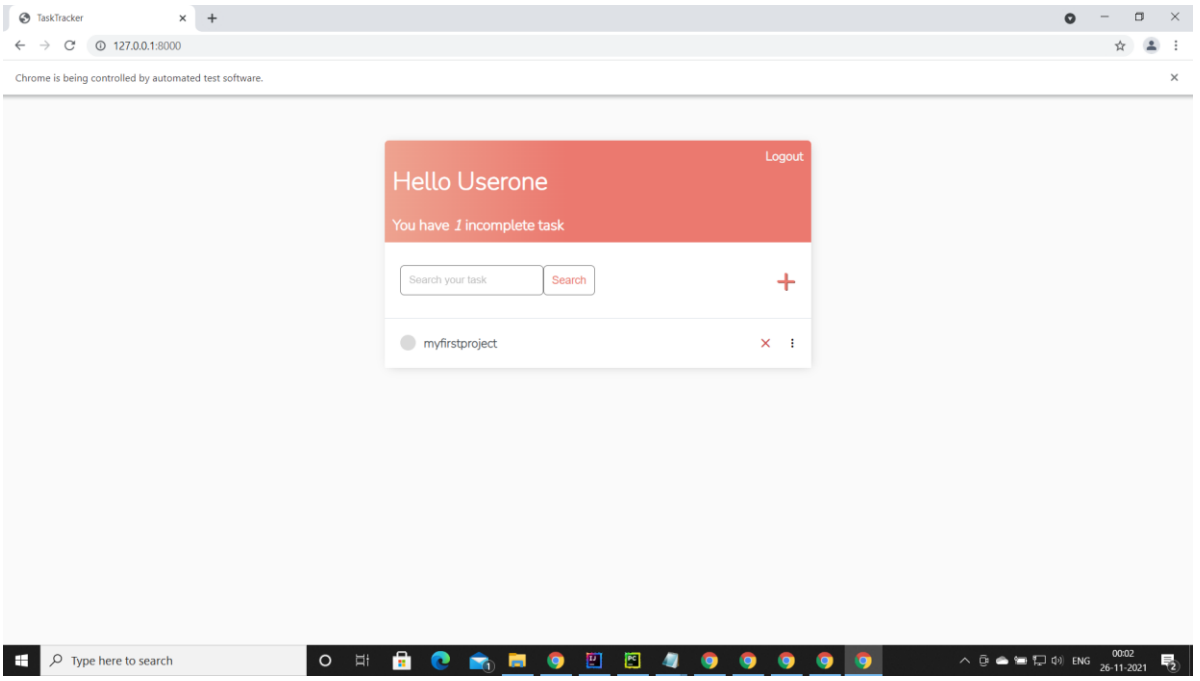

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
SeleniumTraining - Automation.java
SeleniumTraining
Automation
Project
Automation.java
driver.findElement(By.id("id_password"));
WebElement password=driver.findElement(By.id("id_password"));
username.sendKeys(...keysToSend: "demouser");
password.sendKeys(...keysToSend: "demohuman");
String ExpectedUserName = "demouser";
String ActualUserName = "demouser";
Assertions.assertEquals(ExpectedUserName, ActualUserName);
String ExpectedPassword = "demohuman";
String ActualPassword = "demohuman";
Assertions.assertEquals(ExpectedPassword, ActualPassword);
WebElement login = driver.findElement(By.className("button"));
login.click();
System.out.println("Test case Tested Succesfully");
}
public static void main(String[] args) {
// Scanner sc = new Scanner(System.in);
checkTittle();
Run: Automation
ChromeDriver was started successfully.
Nov 25, 2021 10:04:47 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Nov 25, 2021 10:04:47 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 96, so returning the closest version found: 95
Nov 25, 2021 10:04:47 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 96 of 95
Test case Tested Succesfully
Process finished with exit code 0
Type here to search
ENG
22:04
25-11-2021

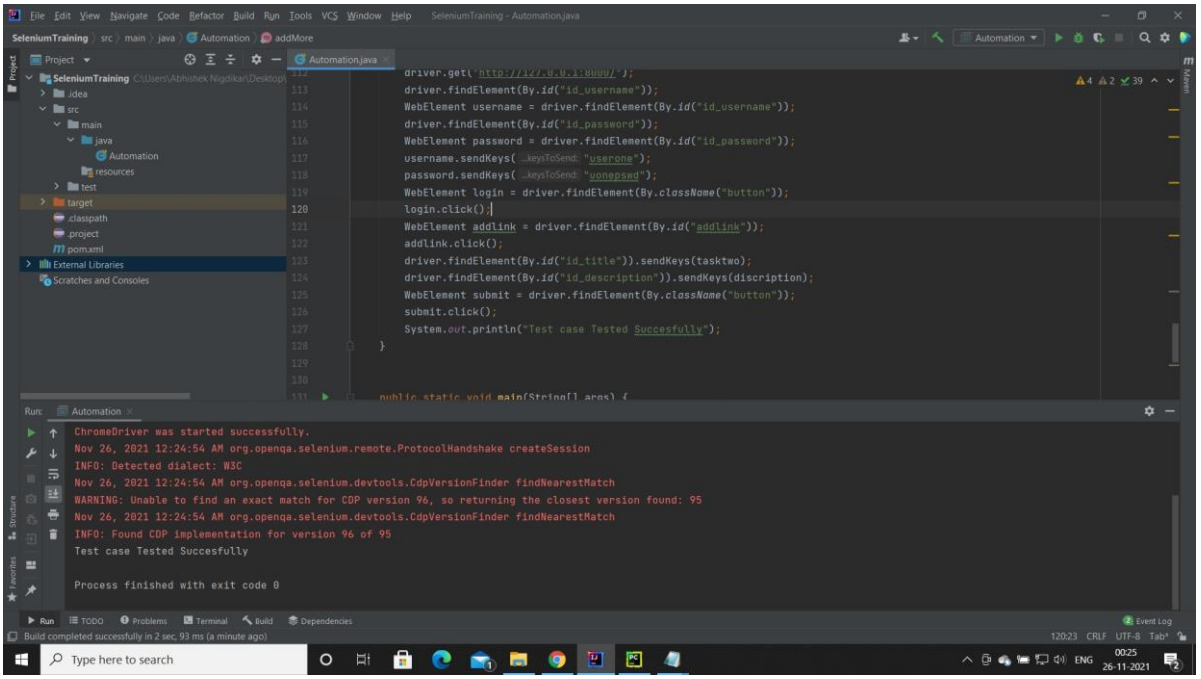

TaskTracker
127.0.0.1:8000
Logout
Hello Demouser
You have 2 incomplete tasks
Search your task
Search
final workdone
final review
Type here to search
ENG
12:53
26-11-2021

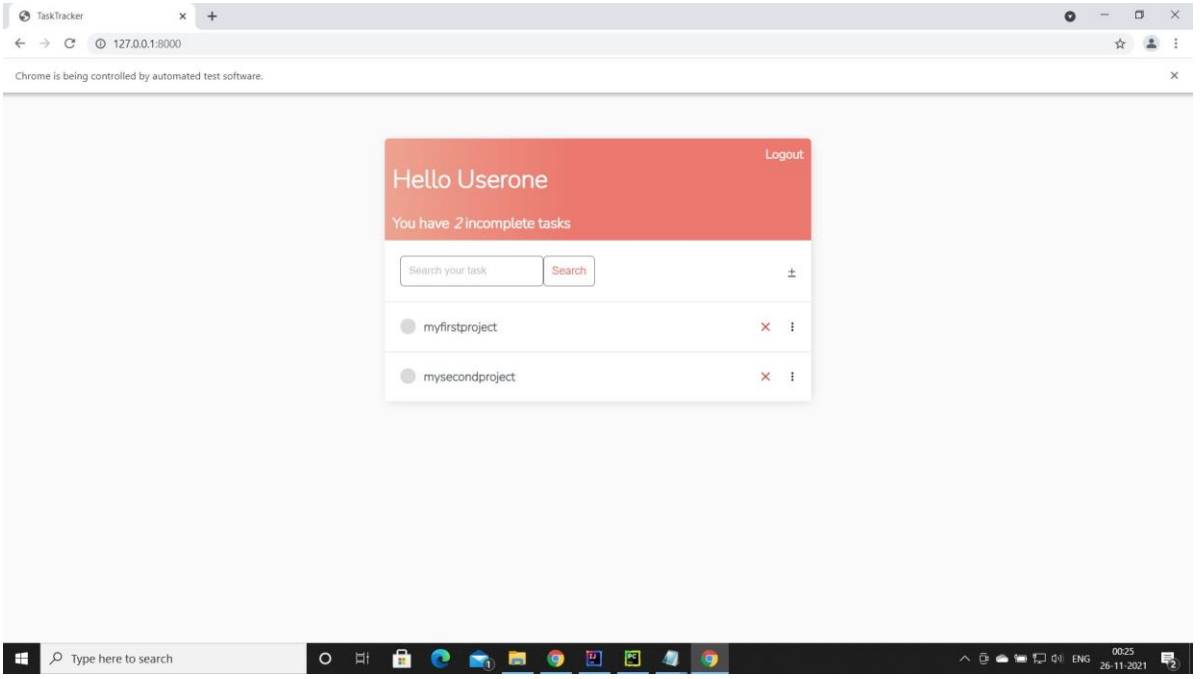

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
SeleniumTraining - Automation.java
SeleniumTraining
Automation
Project
Automation.java
public static void main(String[] args) {
// Scanner sc = new Scanner(System.in);
checkTittle();
login();
register();
}
}
Run: Automation
ChromeDriver was started successfully.
Nov 25, 2021 10:13:12 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Nov 25, 2021 10:13:12 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 96, so returning the closest version found: 95
Nov 25, 2021 10:13:12 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 96 of 95
Test case Tested Succesfully
Process finished with exit code 0
Build completed successfully in 2 sec, 109 ms (moments ago)
Type here to search
ENG
22:13
25-11-2021

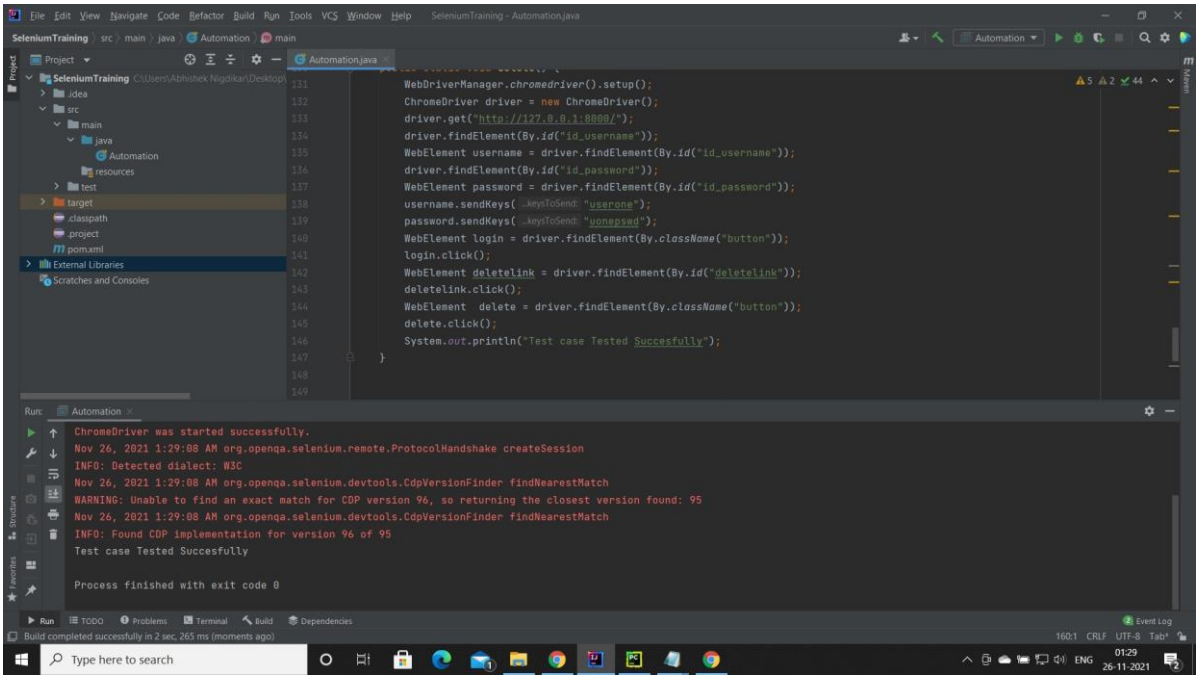

TaskTracker
127.0.0.1:8000
Chrome is being controlled by automated test software.
Logout
Hello Userone
You have 0 incomplete tasks
Search your task
Search
No new tasks are created.
Create a New task !
Type here to search
ENG
22:13
25-11-2021

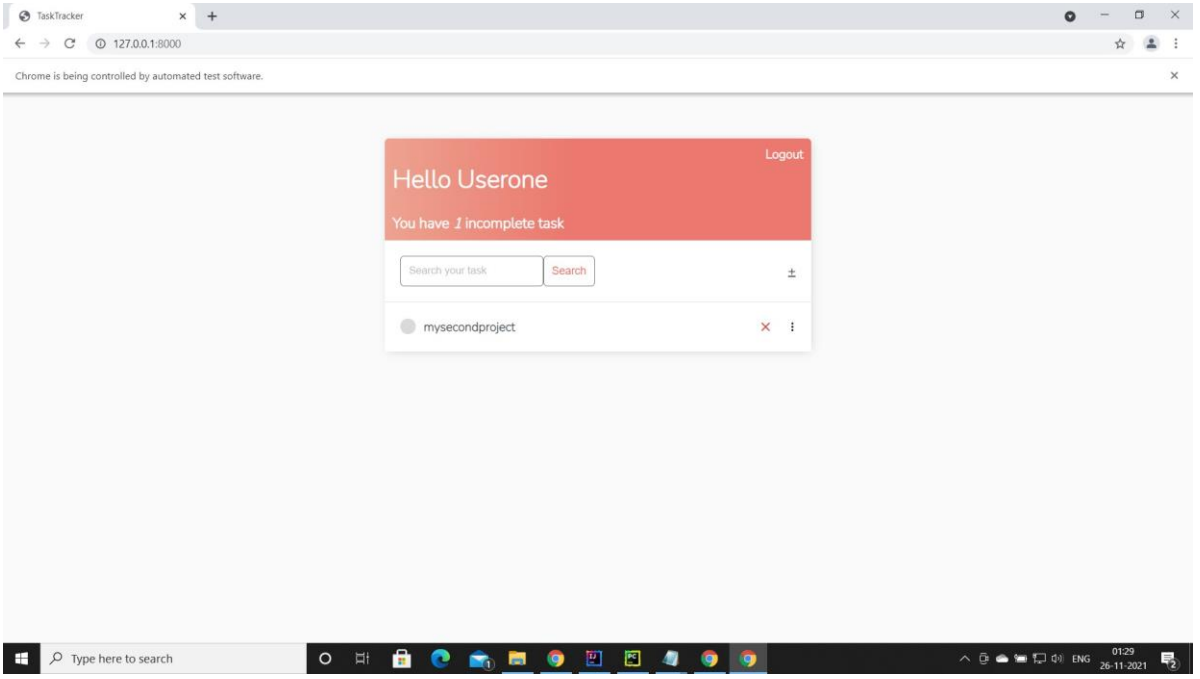

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
SeleniumTraining - Automation.java
SeleniumTraining
Automation
addTask
Project
Automation.java
SeleniumTraining
.idea
src
main
java
Automation
resources
test
target
.classpath
.project
pom.xml
External Libraries
Scratches and Consoles
driver.findElement(By.id("id_username"));
WebElement username = driver.findElement(By.id("id_username"));
driver.findElement(By.id("id_password"));
WebElement password = driver.findElement(By.id("id_password"));
username.sendKeys(...keysToSend: "userone");
password.sendKeys(...keysToSend: "yonepswd");
WebElement login = driver.findElement(By.className("button"));
login.click();
// System.out.println("Test case Tested Succesfully");
WebElement newtask = driver.findElement(By.id("newtask"));
newtask.click();
// WebDriverManager.chromedriver().setup();
driver.findElement(By.id("id_title")).sendKeys(taskone);
driver.findElement(By.id("id_description")).sendKeys(mandatory);
WebElement submit = driver.findElement(By.className("button"));
submit.click();
// WebElement addlink = driver.findElement(By.id("add-link"));
Run: Automation
ChromeDriver was started successfully.
Nov 26, 2021 12:02:24 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Nov 26, 2021 12:02:24 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 96, so returning the closest version found: 95
Nov 26, 2021 12:02:24 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 96 of 95
Test case Tested Succesfully
Process finished with exit code 0
Run TODO Problems Terminal Build Dependencies
Event Log
Build completed successfully in 2 sec, 118 ms (a minute ago)
Type here to search
ENG
00:02
26-11-2021

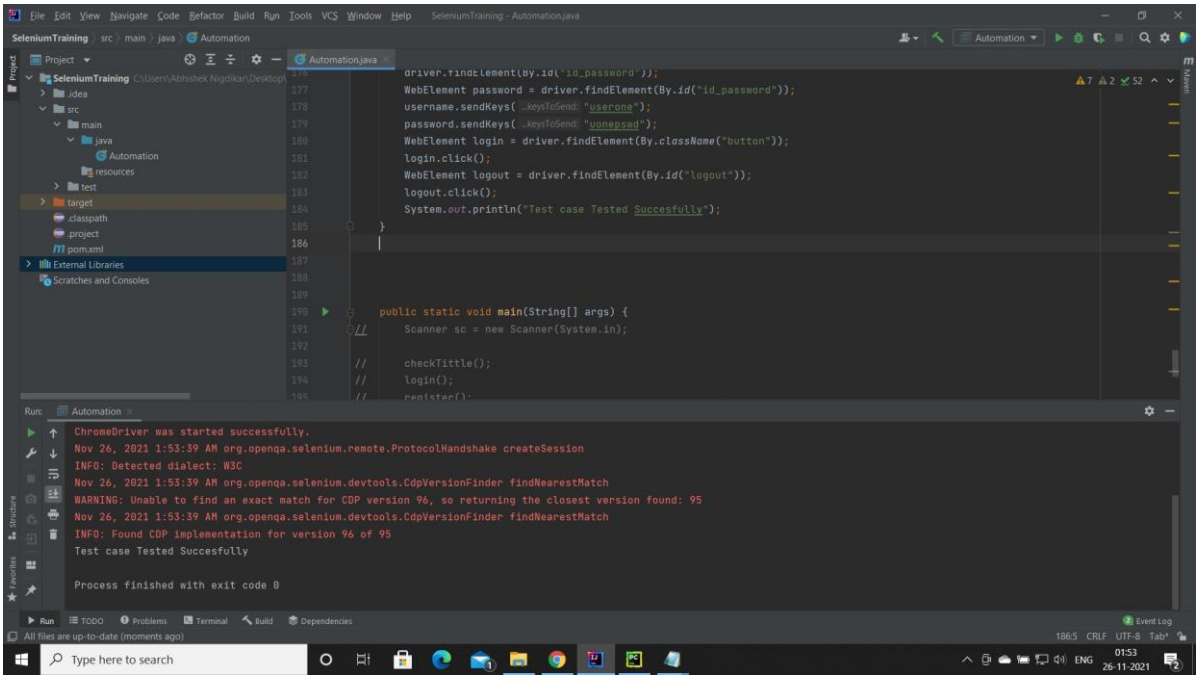

TaskTracker
127.0.0.1:8000
Chrome is being controlled by automated test software.
Logout
Hello Userone
You have 1 incomplete task
Search your task
Search
myfirstproject
Type here to search
ENG
00:02
26-11-2021

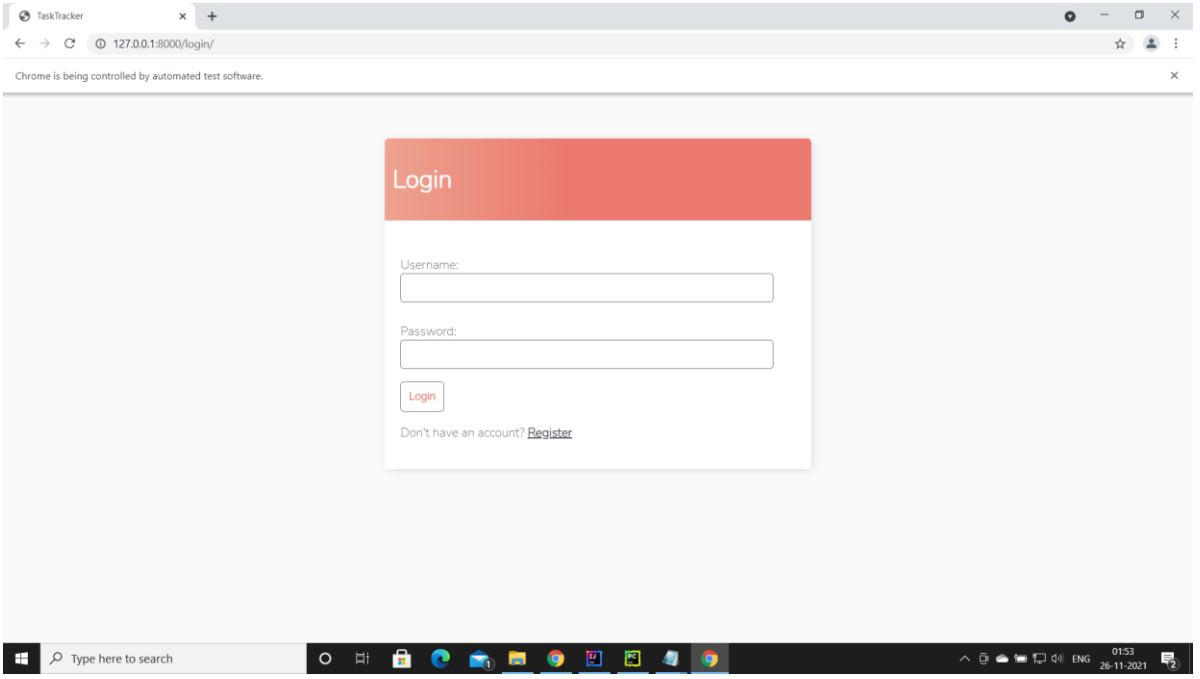

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
SeleniumTraining - Automation.java
SeleniumTraining
Automation
Project
Automation.java
driver.get("http://127.0.0.1:8000/");
driver.findElement(By.id("id_username"));
WebElement username = driver.findElement(By.id("id_username"));
driver.findElement(By.id("id_password"));
WebElement password = driver.findElement(By.id("id_password"));
username.sendKeys(...keysToSend: "userone");
password.sendKeys(...keysToSend: "uonepswd");
WebElement login = driver.findElement(By.className("button"));
login.click();
WebElement addlink = driver.findElement(By.id("addlink"));
addlink.click();
driver.findElement(By.id("id_title")).sendKeys(tasktwo);
driver.findElement(By.id("id_description")).sendKeys(discription);
WebElement submit = driver.findElement(By.className("button"));
submit.click();
System.out.println("Test case Tested Succesfully");
}
public static void main(String[] args) {
Run: Automation
ChromeDriver was started successfully.
Nov 26, 2021 12:24:54 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Nov 26, 2021 12:24:54 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 96, so returning the closest version found: 95
Nov 26, 2021 12:24:54 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 96 of 95
Test case Tested Succesfully
Process finished with exit code 0
Build completed successfully in 2 sec, 93 ms (a minute ago)
Type here to search
ENG
00:25
26-11-2021


TaskTracker
127.0.0.1:8000
Chrome is being controlled by automated test software.
Logout
Hello Userone
You have 2 incomplete tasks
Search your task
Search
myfirstproject
mysecondproject
Type here to search
ENG
00:25
26-11-2021

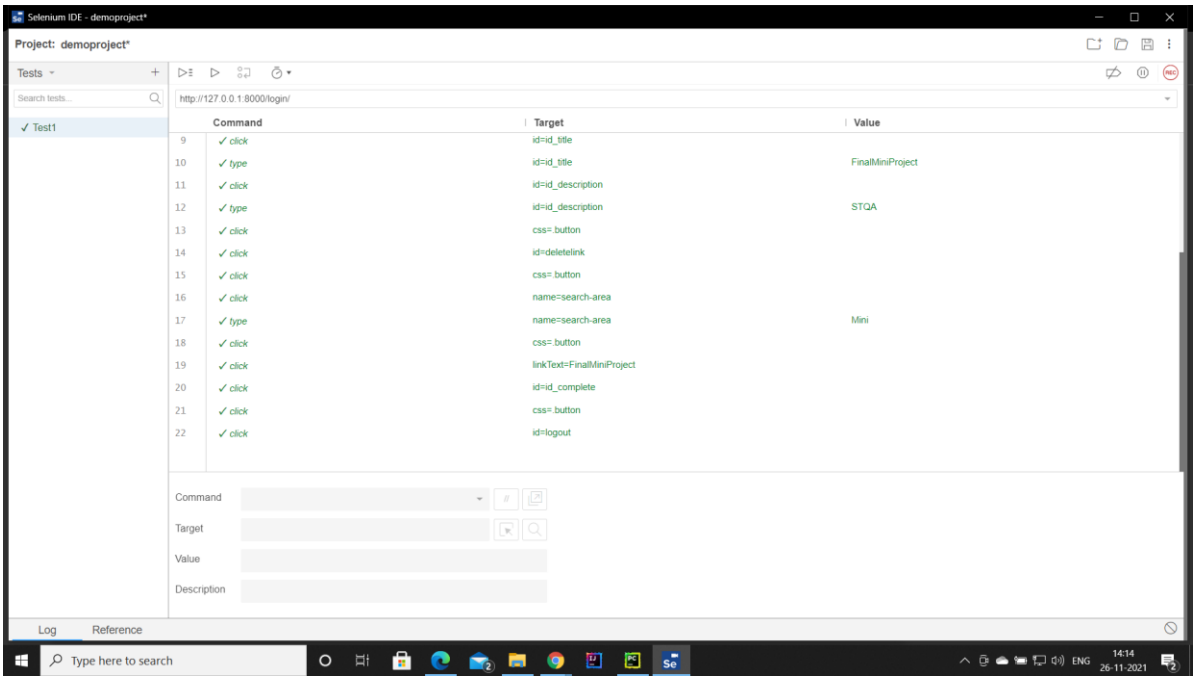

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
SeleniumTraining - Automation.java
SeleniumTraining
Automation
Project
Automation.java
WebDriverManager.chromedriver().setup();
ChromeDriver driver = new ChromeDriver();
driver.get("http://127.0.0.1:8000/");
driver.findElement(By.id("id_username"));
WebElement username = driver.findElement(By.id("id_username"));
driver.findElement(By.id("id_password"));
WebElement password = driver.findElement(By.id("id_password"));
username.sendKeys(...keysToSend: "userone");
password.sendKeys(...keysToSend: "yonepswd");
WebElement login = driver.findElement(By.className("button"));
login.click();
WebElement deletelink = driver.findElement(By.id("deletelink"));
deletelink.click();
WebElement delete = driver.findElement(By.className("button"));
delete.click();
System.out.println("Test case Tested Succesfully");
}
Run: Automation
ChromeDriver was started successfully.
Nov 26, 2021 1:29:08 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Nov 26, 2021 1:29:08 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 96, so returning the closest version found: 95
Nov 26, 2021 1:29:08 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 96 of 95
Test case Tested Succesfully
Process finished with exit code 0
Build completed successfully in 2 sec, 265 ms (moments ago)
Type here to search
ENG
01:29
26-11-2021

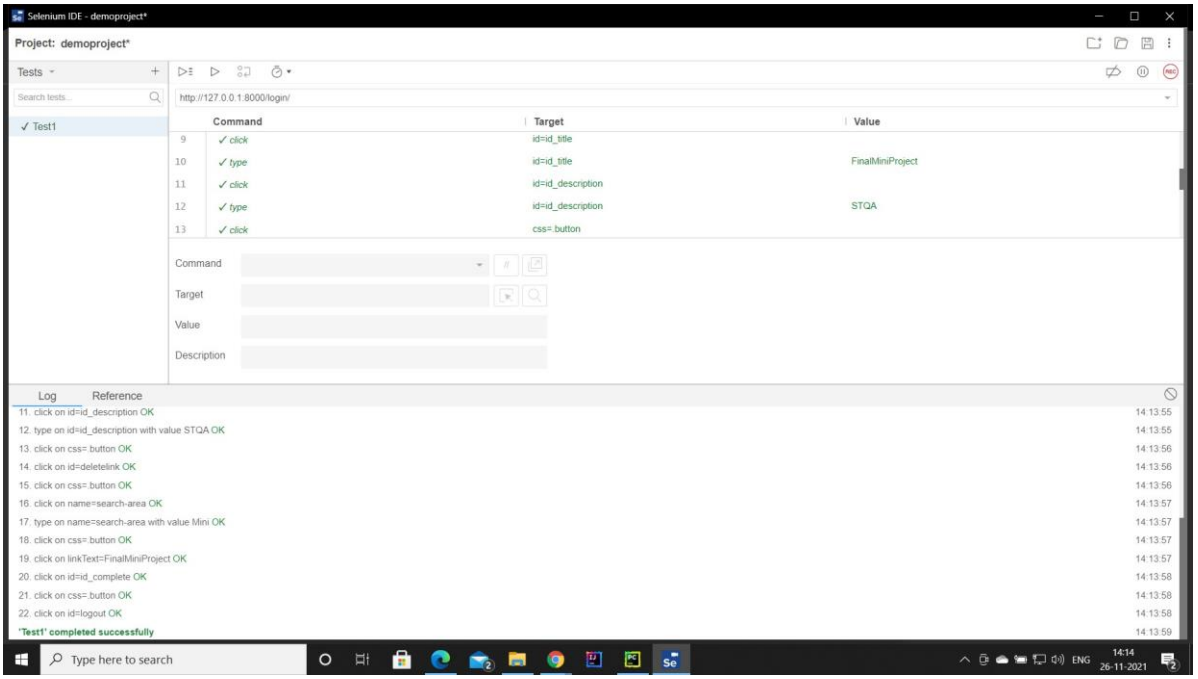

TaskTracker
127.0.0.1:8000
Chrome is being controlled by automated test software.
Logout
Hello Userone
You have 1 incomplete task
Search your task
Search
mysecondproject
Type here to search
ENG
01:29
26-11-2021


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
SeleniumTraining - Automation.java
SeleniumTraining
Automation
Project
Automation.java
driver.findElement(By.id("id_password"));
WebElement password = driver.findElement(By.id("id_password"));
username.sendKeys(...keysToSend: "userone");
password.sendKeys(...keysToSend: "vonepswd");
WebElement login = driver.findElement(By.className("button"));
login.click();
WebElement logout = driver.findElement(By.id("logout"));
logout.click();
System.out.println("Test case Tested Succesfully");
}
public static void main(String[] args) {
// Scanner sc = new Scanner(System.in);
// checkTittle();
// login();
Run: Automation
ChromeDriver was started successfully.
Nov 26, 2021 1:53:39 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Nov 26, 2021 1:53:39 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 96, so returning the closest version found: 95
Nov 26, 2021 1:53:39 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 96 of 95
Test case Tested Succesfully
Process finished with exit code 0
Type here to search
ENG
01:53
26-11-2021


TaskTracker
127.0.0.1:8000/login/
Chrome is being controlled by automated test software.
Login
Username:
Password:
Login
Don't have an account? Register
Type here to search
ENG
01:53
26-11-2021

Selenium IDE :





Selenium Log :

1. open on http://127.0.0.1:8000/login/ OK14:03:23
2.setWindowSize on 1110x824 OK14:03:24
3.click on id=id_username OK14:03:26
4.type on id=id_username with value demouser OK14:03:27
5.click on id=id_password OK14:03:28
6.type on id=id_password with value demohuman OK14:03:30

7.click on css=.button OK14:03:31
8.click on id=addlink OK14:03:32
9.click on id=id_title OK14:03:35
10.type on id=id_title with value FinalMiniProject OK14:03:36
11.click on id=id_description OK14:03:38
12.type on id=id_description with value STQA OK14:03:39
13.click on css=.button OK14:03:40
14.click on id=deletelink OK14:03:42
15.click on css=.button OK14:03:43
16.click on name=search-area OK14:03:45
17.type on name=search-area with value Mini OK14:03:46
18.click on css=.button OK14:03:48
19.click on linkText=FinalMiniProject OK14:03:49
20.click on id=id_complete OK14:03:51
21.click on css=.button OK14:03:52
22.click on id=logout OK14:03:53
'Test1' completed successfully