Experiment No. 1

Aim: Implementation of Django Application with DBsqlite as back-end.

Objective: Development and deployment of Django Application software.

Background:

Django: Django is a high-level web framework for building robust, scalable, and maintainable web applications using the Python programming language. It follows the Model-View-Controller (MVC) architectural pattern and encourages the "Don't Repeat Yourself" (DRY) principle.

Key features of Django include an Object-Relational Mapping (ORM) system for database interaction, a dynamic URL routing system, and a templating engine for building dynamic and data-driven web pages. It also provides a built-in administration panel for managing site content.

Django promotes rapid development by emphasizing convention over configuration, providing a secure and scalable foundation for web applications. It includes features like authentication, authorization, and built-in protection against common web vulnerabilities.

Django's modularity allows developers to use specific components as needed, and its extensive documentation and active community make it a popular choice for web development. It supports various databases, including PostgreSQL, MySQL, and SQLite.

Additionally, Django REST framework extends the framework to facilitate the creation of RESTful APIs, making it versatile for both traditional web applications and modern web services. Overall, Django is known for its simplicity, versatility, and efficiency in building complex web applications.

SQlite:

SQLite is a lightweight, serverless, self-contained, and open-source relational database management system. It is particularly well-suited for embedded systems and smaller-scale applications due to its simplicity and minimal configuration requirements.

SQLite databases are stored in a single disk file, making them portable and easy to manage. It supports standard SQL features, transactions, and provides ACID (Atomicity, Consistency, Isolation, Durability) compliance. Unlike client-server databases, SQLite is embedded directly into the application, eliminating the need for a separate database server process.

Developers often choose SQLite for projects with modest data storage requirements, such as mobile applications, desktop applications, and small to medium-sized web applications. Its low setup overhead and zero configuration make it a popular choice for prototyping and development.

Despite its lightweight nature, SQLite supports various data types, indexing, and complex queries. It is compatible with many programming languages, including Python, which integrates well with the Django web framework. Django can utilize SQLite as its default database backend during development, offering a convenient and easy-to-use option for beginners or projects with less demanding database needs

Django and SQLite:

Django, a Python web framework, integrates seamlessly with SQLite, a lightweight relational database. Developers can initiate a Django project with SQLite as the default database using commands like django-admin startproject projectname and configure database settings in settings.py. This combination provides a quick, easy-to-use solution for developing web applications with built-in admin functionality and minimal setup.

Pre-Lab Task: Setting Up the Environment

Objective:
The pre-lab task aims to ensure that your development environment is properly configured for the upcoming Django and SQLite lab. This includes installing necessary software and verifying the setup.

Requirements:

Python installed on your machine.
A code editor (e.g., Visual Studio Code).
A terminal or command prompt.

Task Steps:

Install Python:
Ensure that Python is installed on your machine. If not, download and install it from the official Python website: Python Downloads

Install a Code Editor:
Choose and install a code editor suitable for Python development. Visual Studio Code is recommended and can be downloaded from: Visual Studio Code

Verify Python Installation:
Open a terminal or command prompt and check if Python is installed by running the appropriate command for your system:

On Windows: python --version
On Unix or MacOS: python3 --version
Ensure that the version displayed matches the one you installed.

Install Django:
Install Django, a Python web framework, by using the package manager pip. You can do this by running the following command in the terminal:

bash
Copy code
pip install django
Verify the installation by checking the Django version:

bash
Copy code django-
admin --version
Create a Test Django Project:
Create a test Django project to ensure that the framework is set up correctly. This involves using the django-admin command to start a new project and running the development server. Verify that you can see the Django welcome page in your web browser.

Understand the Basics:

Familiarize yourself with basic Django concepts, such as models, views, and templates. You can refer to the official Django documentation, which provides comprehensive guides and tutorials: Django Documentation

Conclusion:
You have completed the pre-lab tasks to set up your development environment for the upcoming Django and SQLite lab. Ensure that Python, the code editor, and Django are installed and working correctly before proceeding to the lab task.

INLAB:
Signup.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Modern Signup Page</title>
    <link rel="stylesheet" href="/static/css/signup.css"> <!-- Link the external CSS file -->
</head>
<body>
    <div class="container">
        <h2> Signup</h2>
        <div class="decoration"></div>
        <form method='POST'>
            {% csrf_token %}
            <div class="input-group">
                <label for="username">Username</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="input-group">
                <label for="email">Email</label>
                <input type="email" id="email" name="email" required>
            </div>
            <div class="input-group">
                <label for="password">Password</label>
                <input type="password" id="password" name="password" required>
            </div>
            <div class="input-group">
                <label for="confirm-password">Confirm Password</label>
                <input type="password" id="confirm-password" name="confirm-password" required>
            </div>
            <button type="submit" class="btn-primary">Signup</button>
        </form>
        <p>Already have an account? <a href="/login" class="btn-secondary">Login</a></p>
    </div>
</body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
    <link rel="stylesheet" href="/static/css/signup.css"> <!-- Link the external CSS file -->
</head>
<body>
    <div class="container">
        <h2>Login</h2>
        <div class="decoration"></div>
        <form method="POST">
            {% csrf_token %}
            <div class="input-group">
                <label for="username">Username</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="input-group">
                <label for="password">Password</label>
                <input type="password" id="password" name="pass" required>
            </div>
            <button type="submit" class="btn-primary">Login</button>
        </form>
        <p>Don't have an account? <a href="/" class="btn-secondary">Create an Account</a></p>
    </div>
</body>
</html>
```

Style.CSS

```
/* styles.css */
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color: #090909f2;
    margin: 0;
    padding: 0;
    display: flex; justify-
    content: center; align-
    items: center; min-
    height: 100vh;
    color: #fff;
}

.container {
    width: 400px;
    background: #494846c3;
    border-radius: 10px;
```

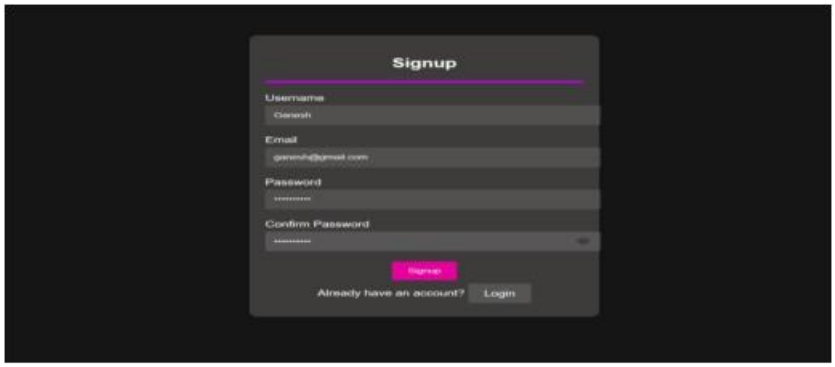
```
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
    padding: 20px;
    text-align: center;
}

.container h2 {
    color: #fff;
}

.input-group {
    margin-bottom: 20px;
}

.input-group label {
    display: block;
    text-align: left;
    margin-bottom: 5px;
    color: #fff;
}

.input-group input {
    width: 100%;
    padding: 10px;
    border: 1px solid #555;
    border-radius: 3px;
    outline: none;
    background: rgba(255, 255, 255, 0.1);
    color: #fff;
}

.input-group input:focus {
    background: rgba(228, 222, 222, 0.16);
}

.btn-primary {
    background: #e6009d;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

.btn-secondary {
    background: #555;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 3px;
    cursor: pointer;
    text-decoration: none;
```

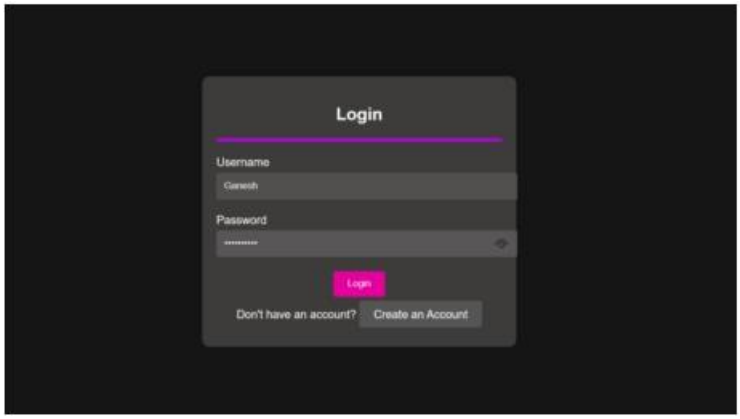
```
}

.btn-secondary:hover {
    background: #444;
}

/* Add some decorative elements */
.decoration {
    background: #b703df;
    height: 5px;
    margin: 20px 0;
}
```

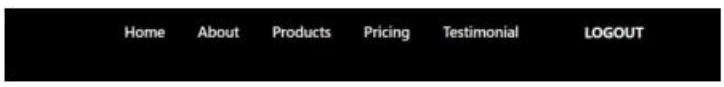
Views.py

```
from django.shortcuts import render,redirect
from django.http import HttpResponse
from django.contrib.auth.models import User
from django.contrib.auth import authenticate,login,logout
from django.contrib.auth.decorators import login_required


@login_required(login_url="login")
def home(request):
    data=index_info.objects.all()
    dict={
        'info':data,
    }
    return render(request,'index.html',dict)

def signup(request):
    if request.method=="POST":
        uname=request.POST.get('username')
        email=request.POST.get('email')
        pass1=request.POST.get('password')
        pass2=request.POST.get('confirm-password')

        if pass1!=pass2:
            return HttpResponse("password and confirm-password is not matching, Please
recheck!")
        else:
            my_user=User.objects.create_user(uname,email,pass1)
            my_user.save()
            return redirect("login")


    return render(request,'signup.html')

def loginPage(request):
    if request.method=="POST":
        username=request.POST.get('username')
```

```
        pass1=request.POST.get('pass')


        user=authenticate(request,username=username,password=pass1)

        if user is not None:
            login(request,user)
            return redirect('home')
        else:
            return HttpResponse("username or password is incorrect")


    return render(request,"login.html")


def LogoutPage(request):
    logout(request)
    return redirect("login")
```







username or password is incorrect

PostLab:

Performed Login and registration application using Django and SQLite.

Experiment No. 2

Aim: Study and perform Data Analytics and Data Visualization using Numpy, Pandas and Matplotlib

1. Objective: To perform data analysis and visualization on a sample dataset .

2. Background : NumPy

It is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionality. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

Operations using NumPy

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.

- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

NumPy – A Replacement for MatLab

NumPy is often used along with packages like **SciPy** (Scientific Python) and **Mat–plotlib** (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing. However, Python alternative to MatLab is now seen as a more modern and complete programming language. It is open source, which is an added advantage of NumPy.

Pandas

Pandas is an open-source Python Library used for high-performance data manipulation and data analysis using its powerful data structures. Python with pandas is in use in a variety of academic and commercial domains, including Finance, Economics, Statistics, Advertising, Web Analytics, and more. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, organize, manipulate, model, and analyse the data.

Below are the some of the important features of Pandas which is used specifically for Data processing and Data analysis work.

Key Features of Pandas

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.

- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Pandas deals with the following three data structures −

- **Series**
- **DataFrame**

These data structures are built on top of Numpy array, making them fast and efficient.

Data Structure	Dimensions	Description
Series	1	1D labeled homogeneous array, size-immutable.
Data Frames	2	General 2D labeled, size-mutable tabular structure with potentially heterogeneously typed columns.

DataFrame is widely used and it is the most important data structures.

Series

Series is a one-dimensional array like structure with homogeneous data. For example, the following series is a collection of integers 10, 23, 56, …

Key Points of Series

- Homogeneous data
- Size Immutable
- Values of Data Mutable

DataFrame

DataFrame is a two-dimensional array with heterogeneous data. For example,

Name	Age	Gender	Rating
Steve	32	Male	3.45
Lia	28	Female	4.6
Vin	45	Male	3.9

Katie	38	Female	2.78

The table represents the data of a sales team of an organization with their overall performance rating. The data is represented in rows and columns. Each column represents an attribute and each row represents a person.

Key Points of Data Frame

- Heterogeneous data
- Size Mutable
- Data Mutable

Matplotlib

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Matplotlib has a procedural interface named the Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by MathWorks. Matplotlib along with NumPy can be considered as the open source equivalent of MATLAB.

Matplotlib was originally written by John D. Hunter in 2003. The current stable version is 2.2.0 released in January 2018.

3. Pre-lab Task:

1. Installation of Ubuntu 18.04
2. Installation of Anaconda 3 with Jupyter Notebook
3. Installation of Numpy Package
4. Installation of Pandas Package
5. Installation of Matplotlib Package

4. In Lab Tasks

Perform the following data analysis and visualization on the given dataset.

Write a Python program to display first 5 rows from COVID-19 dataset. Also print the dataset information and check the missing

Solution :

```
import pandas as pd

covid_data=pd.read_csv('covid_19_data.csv')
print("\nDataset First Five Rows:")
print(covid_data.head(5))

print("\nDataset Information :")
print(covid_data.info())
```

```
print("\nMissing data information:")
print(covid_data.isna().sum())
```

Output :

```
Dataset First Five Rows:
   SNo ObservationDate Province/State Country/Region       Last Update \
0    1      01/22/2020          Anhui Mainland China 1/22/2020 17:00
1    2      01/22/2020        Beijing Mainland China 1/22/2020 17:00
2    3      01/22/2020      Chongqing Mainland China 1/22/2020 17:00
3    4      01/22/2020         Fujian Mainland China 1/22/2020 17:00
4    5      01/22/2020          Gansu Mainland China 1/22/2020 17:00

   Confirmed  Deaths  Recovered
0        1.0     0.0        0.0
1       14.0     0.0        0.0
2        6.0     0.0        0.0
3        1.0     0.0        0.0
4        0.0     0.0        0.0

Dataset Information :

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156292 entries, 0 to 156291
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   SNo             156292 non-null int64
 1   ObservationDate 156292 non-null object
 2   Province/State  111979 non-null object
 3   Country/Region  156292 non-null object
 4   Last Update     156292 non-null object
 5   Confirmed       156292 non-null float64
 6   Deaths          156292 non-null float64
 7   Recovered       156292 non-null float64
dtypes: float64(3), int64(1), object(4)
memory usage: 9.5+ MB
None

Missing data information:
SNo                0
ObservationDate    0
Province/State     44313
Country/Region     0
Last Update        0
Confirmed          0
Deaths             0
```

```
Recovered          0
dtype: int64
```

Write a Python program to get the latest number of confirmed, deaths, recovered and active cases of Novel Coronavirus (COVID-19) Country wise.

Solution:

```
import pandas as pd
covid_data= pd.read_csv('covid_19_data.csv')
covid_data['Active'] = covid_data['Confirmed'] - covid_data['Deaths'] - covid_data['Recovered']


result = covid_data.groupby('Country/Region')[['Confirmed', 'Deaths', 'Recovered',
'Active']].sum().reset_index()
print(result)
```

Output:

```
    Country/Region Confirmed     Deaths Recovered      Active
0                Azerbaijan        1.0       0.0        0.0        1.0
1              ('St. Martin',)     2.0       0.0        0.0        2.0
2               Afghanistan 6289387.0 208903.0 4251819.0 1828665.0
3                   Albania 1685359.0 45778.0  937725.0  701856.0
4                   Algeria 6560551.0 262137.0 4499006.0 1799408.0
..                      ...       ...       ...       ...       ...
221          Western Sahara     2011.0     174.0     1536.0      301.0
222                   Yemen 286662.0 81116.0     157104.0    48442.0
223                  Zambia 1592737.0 36186.0    1431407.0 125144.0
224                Zimbabwe  829416.0  22839.0    628780.0 177797.0
225 occupied Palestinian territory       25.0       0.0        0.0       25.0

[226 rows x 5 columns]
```

Write a Python program to get the latest number of confirmed deaths and recovered people of Novel Coronavirus (COVID-19) cases Country/Region - Province/State wise.

Solution:

```
import pandas as pd
covid_data= pd.read_csv('covid_19_data.csv')
data = covid_data.groupby(['Country/Region', 'Province/State'])[['Confirmed', 'Deaths',
'Recovered']].max()
pd.set_option('display.max_rows', None)
print(data)
```

Output:

Country/Region	Province/State	Confirmed \
Australia	Australian Capital Territory	114.0
	Diamond Princess cruise ship	0.0
	External territories	0.0
	From Diamond Princess	8.0
	Jervis Bay Territory	0.0
	New South Wales	4498.0
	Northern Territory	46.0
	Queensland	1185.0
	South Australia	544.0
	Tasmania	231.0
	Victoria	20347.0
	Western Australia	787.0
Austria	None	2.0

Write a Python program to get the Chinese province wise cases of confirmed, deaths and recovered cases of Novel Coronavirus (COVID-19).

Solution:

```
import pandas as pd

covid_data= pd.read_csv('covid_19_data.csv')

c_data = covid_data[covid_data['Country/Region']=='Mainland China']
c_data = c_data[['Province/State', 'Confirmed', 'Deaths', 'Recovered']]
result = c_data.sort_values(by='Confirmed', ascending=False)

result = result.reset_index(drop=True)
print(result)
```

Output:

	Province/State	Confirmed	Deaths	Recovered
0	Hubei	68148.0	4512.0	63627.0
1	Hubei	68148.0	4512.0	63627.0
2	Hubei	68147.0	4512.0	63627.0
3	Hubei	68147.0	4512.0	63627.0
4	Hubei	68147.0	4512.0	63627.0
...	...	...	...	...
9262	Gansu	0.0	0.0	0.0
9263	Jilin	0.0	0.0	0.0
9264	Heilongjiang	0.0	0.0	0.0
9265	Inner Mongolia	0.0	0.0	0.0
9266	Xinjiang	0.0	0.0	0.0

[9267 rows x 4 columns]

Write a Python program to list countries with no cases of Novel Coronavirus (COVID-19) recovered.

Solution:

```
import pandas as pd
covid_data= pd.read_csv('covid_19_data.csv')
data = covid_data.groupby('Country/Region')[['Confirmed', 'Deaths', 'Recovered']].sum().reset_index()
result = data[data['Recovered']==0][['Country/Region', 'Confirmed', 'Deaths', 'Recovered']]
print(result)
```

Output:

	Country/Region	Confirmed	Deaths	Recovered
0	Azerbaijan	1.0	0.0	0.0
1	('St. Martin',)	2.0	0.0	0.0
10	Aruba	19.0	0.0	0.0
15	Bahamas, The	10.0	0.0	0.0
37	Cape Verde	1.0	0.0	0 0
38	Cayman Islands	3.0	0.0	0.0
41	Channel Islands	1.0	0.0	0.0
50	Curacao	2.0	0.0	0.0
58	East Timor	1.0	0.0	0.0
67	Faroe Islands	10.0	0.0	0 0
74	Gambia, The	4.0	0.0	0 0
80	Greenland	3.0	0.0	0.0
82	Guadeloupe	187.0	0.0	0.0
83	Guam	6.0	0.0	0.0
85	Guernsey	3.0	0.0	0.0
105	Jersey	6.0	0.0	0.0
121	MS Zaandam	2069.0	458.0	0.0
131	Martinique	172.0	6.0	0.0
134	Mayotte	21.0	0.0	0.0
149	North Ireland	1.0	0.0	0.0

155	Palestine	86.0	0.0	0.0
163	Puerto Rico	3.0	0.0	0.0
165	Republic of Ireland	21.0	0.0	0.0
166	Republic of the Congo	1.0	0.0	0.0
167	Reunion	137.0	0.0	0.0
171	Saint Barthelemy	17.0	0.0	0.0
192	St. Martin	2.0	0.0	0.0
202	The Bahamas	3.0	0.0	0.0
203	The Gambia	1.0	0.0	0.0
216	Vanuatu	6.0	0.0	0.0
217	Vatican City	4.0	0.0	0.0
225	occupied Palestinian territory	25.0	0.0	0.0

Write a Python program to get the top 10 countries data (Last Update, Country/Region, Confirmed, Death s, Recovered) of Novel Coronavirus (COVID-19).

Solution:

```
import pandas as pd
covid_data= pd.read_csv('covid_19_data.csv', usecols = ['Last Update', 'Country/Region',
'Confirmed', 'Deaths', 'Recovered'])
result = covid_data.groupby('Country/Region').max().sort_values(by='Confirmed',
ascending=False)[:10]
pd.set_option('display.max_column', None)
print(result)
```

Output:

```
              Last Update Confirmed Deaths Recovered
Country/Region
France         4/6/20 9:37 1867721.0 42215.0  113017.0
India          4/6/20 9:37 1747242.0 45974.0 1615379.0
Argentina      4/6/20 9:37 1310491.0 35436.0 1129102.0
UK             4/6/20 9:37 1174979.0 45592.0     816.0
Brazil         4/6/20 9:37 1168640.0 40564.0 1051778.0
US             4/6/20 9:37 1059753.0 34032.0 4174884.0
Iran           4/6/20 9:37 762068.0 41493.0  558818.0
South Africa   4/6/20 9:37 751024.0 20241.0  693467.0
Poland         4/6/20 9:37 712972.0 10348.0  294783.0
Iraq           4/6/20 9:37 519152.0 11670.0  447039.0
```

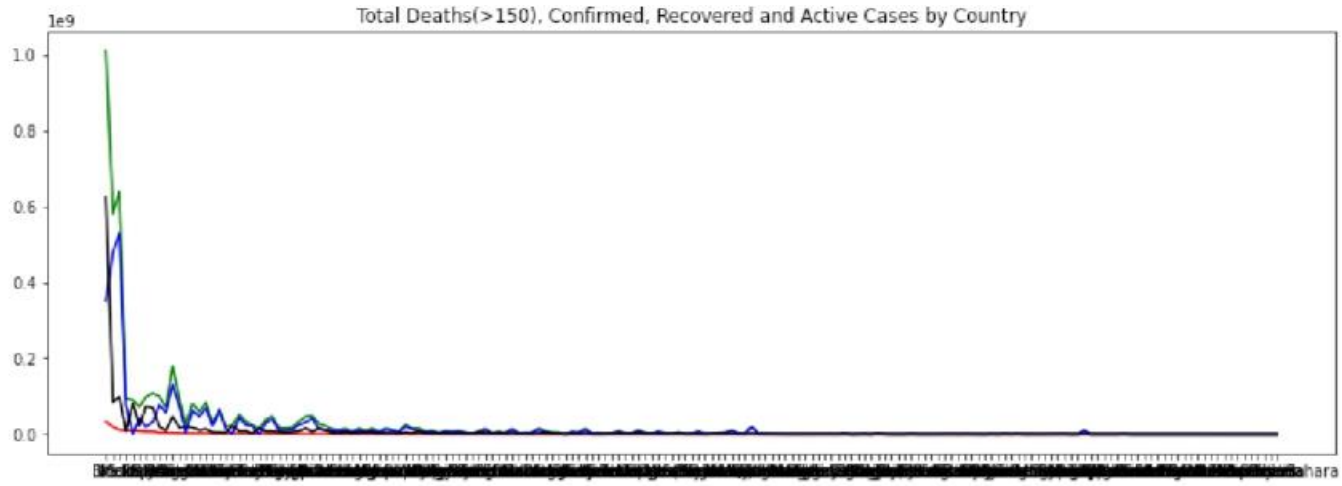
Write a Python program to create a plot (lines) of total deaths, confirmed, recovered and active cases Country wise where deaths greater than 150.

Solution:

```
import pandas as pd
import matplotlib.pyplot as plt
covid_data= pd.read_csv('covid_19_data.csv', usecols = ['Last Update', 'Country/Region', 'Confirmed',
'Deaths', 'Recovered'])
covid_data['Active'] = covid_data['Confirmed'] - covid_data['Deaths'] - covid_data['Recovered']

r_data = covid_data.groupby(["Country/Region"])[["Deaths", "Confirmed", "Recovered",
"Active"]].sum().reset_index()
```

```
r_data = r_data.sort_values(by='Deaths', ascending=False)
r_data = r_data[r_data['Deaths']>50]
plt.figure(figsize=(15, 5))
plt.plot(r_data['Country/Region'], r_data['Deaths'],color='red')
plt.plot(r_data['Country/Region'], r_data['Confirmed'],color='green')
plt.plot(r_data['Country/Region'], r_data['Recovered'], color='blue')
plt.plot(r_data['Country/Region'], r_data['Active'], color='black')

plt.title('Total Deaths(>150), Confirmed, Recovered and Active Cases by Country')
plt.show()
```

Output:

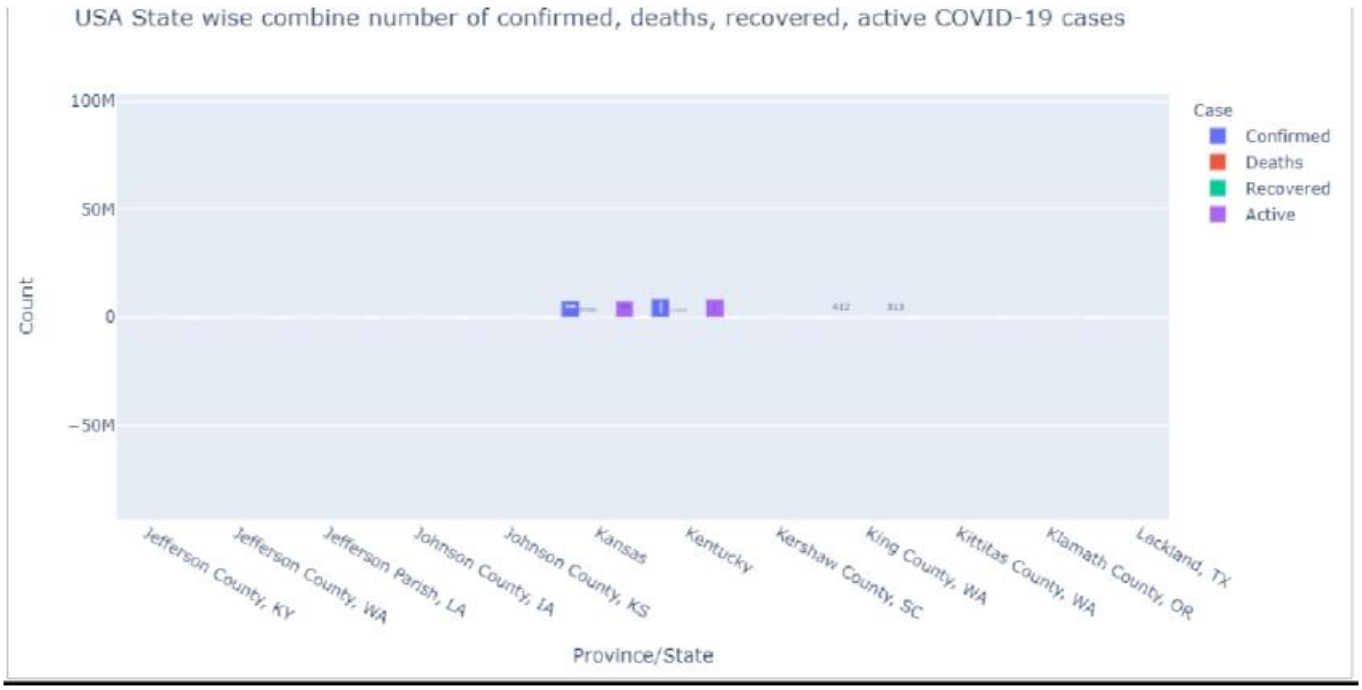


Write a Python program to visualize the state/province wise combine number of confirmed, deaths, recovered, active Novel Coronavirus (COVID-19) cases in USA.

Solution:

```
import pandas as pd
import plotly.express as px
covid_data= pd.read_csv('covid_19_data.csv')
covid_data['Active'] = covid_data['Confirmed'] - covid_data['Deaths'] - covid_data['Recovered']
combine_us_data = covid_data[covid_data['Country/Region']=='US'].drop(['Country/Region'], axis=1)
combine_us_data = combine_us_data[combine_us_data.sum(axis = 1) > 0]
combine_us_data = combine_us_data.groupby(['Province/State'])[['Confirmed', 'Deaths', 'Recovered',
'Active']].sum().reset_index()
combine_us_data = pd.melt(combine_us_data, id_vars='Province/State', value_vars=['Confirmed',
'Deaths', 'Recovered', 'Active'], value_name='Count', var_name='Case')
fig = px.bar(combine_us_data, x='Province/State', y='Count', text='Count', barmode='group',
color='Case', title='USA State wise combine number of confirmed, deaths, recovered, active COVID-19
cases')
fig.show()
```

Output:

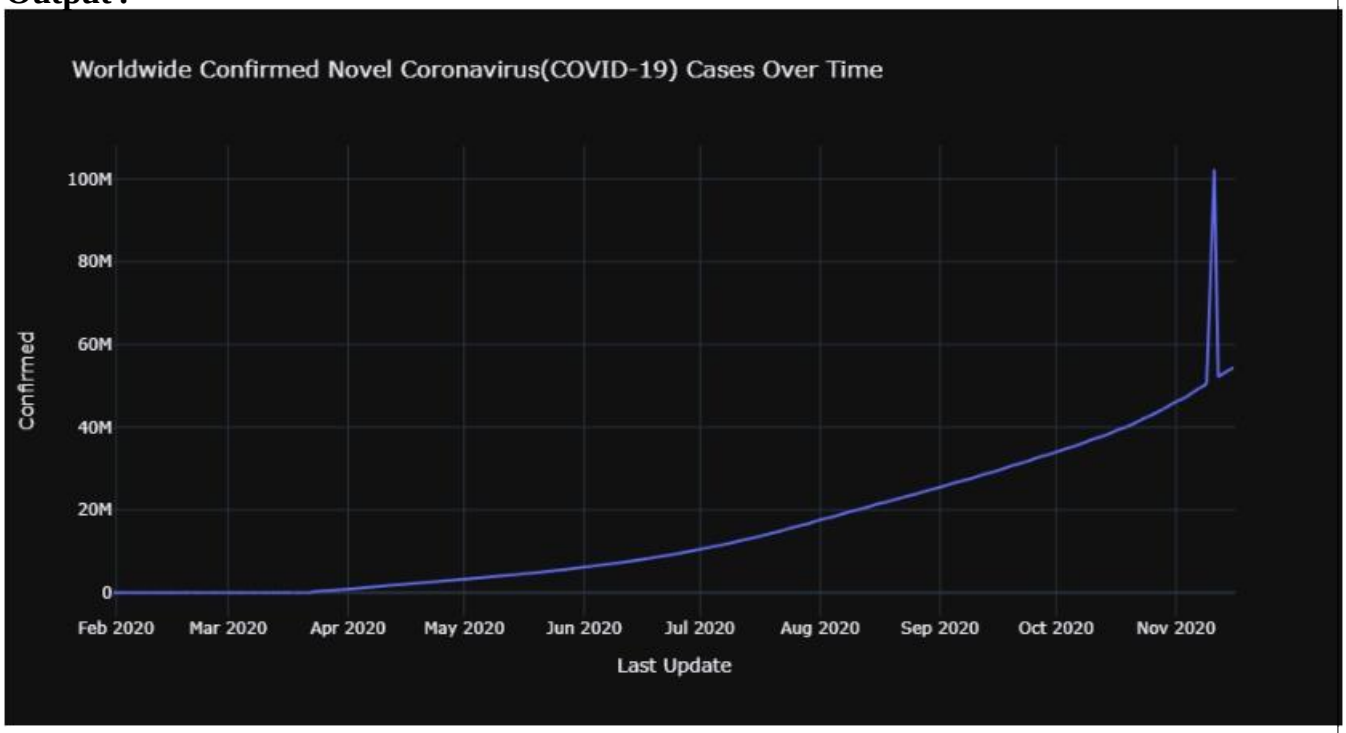


Write a Python program to visualize Worldwide Confirmed Novel Coronavirus (COVID-19) cases over time.

Solution:

```
import pandas as pd
import plotly.express as px
import plotly.io as pio

pio.templates.default = "plotly_dark"


covid_data= pd.read_csv('covid_19_data.csv')

grouped = covid_data.groupby('Last Update')[['Last Update', 'Confirmed',
'Deaths']].sum().reset_index()

fig = px.line(grouped, x="Last Update", y="Confirmed",

        title="Worldwide Confirmed Novel Coronavirus(COVID-19) Cases Over Time") fig.show()
```

Output :



5. Post Lab Tasks

Performed data analysis and visualization on the sample dataset using Numpy, Pandas and Matplotlib

Experiment No. 3

Aim: Implementation of ASP.net Application with Sql Server as back-end .

1. Objective: Development and deployment of ASP.net Application software.

2. Background:

ASP.NET is an open-source,[2] server-side web-application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, applications and services.

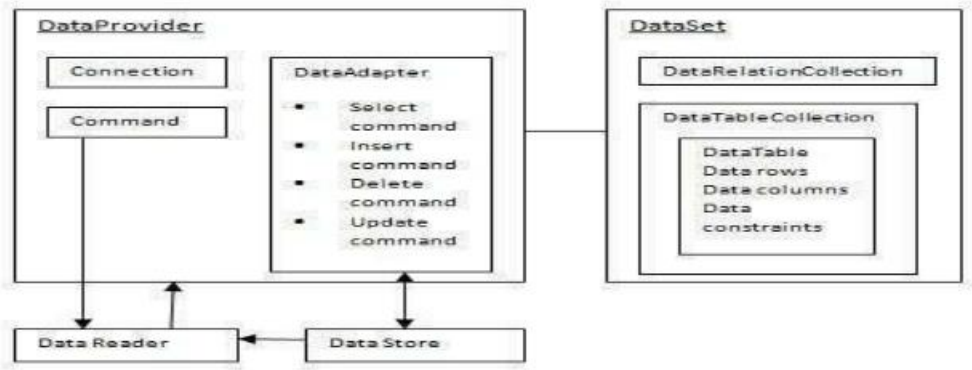
It was first released in January 2002 with version 1.0 of the .NET Framework and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. TheASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

ASP.NET's successor is ASP.NET Core. It is a re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The new framework uses the new open-source .NET Compiler Platform (codename "Roslyn") and is cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) have merged into a unified MVC

ASP.net and SqlServer (ADO.net)

ADO.NET provides a bridge between the front end controls and the back end database. The ADO.NET objects encapsulate all the data access operations and the controls interact with these objects to display data, thus hiding the details of movement of data.

The following figure shows the ADO.NET objects at a glance:



The DataSet Class

The dataset represents a subset of the database. It does not have a continuous connection to the database. To update the database a reconnection is required. The DataSet contains DataTable objects and DataRelation objects. The DataRelation objects represent the relationship between two tables.

Following table shows some important properties of the DataSet class:

Properties	**Description**
CaseSensitive	Indicates whether string comparisons within the data tables are case-sensitive.
Container	Gets the container for the component.
DataSetName	Gets or sets the name of the current data set.
DefaultViewManager	Returns a view of data in the data set.
DesignMode	Indicates whether the component is currently in design mode.
EnforceConstraints	Indicates whether constraint rules are followed when attempting any update operation.
Events	Gets the list of event handlers that are attached to this component.
ExtendedProperties	Gets the collection of customized user information associated with the DataSet.
HasErrors	Indicates if there are any errors.
IsInitialized	Indicates whether the DataSet is initialized.

Locale	Gets or sets the locale information used to compare strings within the table.
Namespace	Gets or sets the namespace of the DataSet.
Prefix	Gets or sets an XML prefix that aliases the namespace of the DataSet.
Relations	Returns the collection of DataRelation objects.
Tables	Returns the collection of DataTable objects.

The following table shows some important methods of the DataSet class:

Methods	**Description**
AcceptChanges	Accepts all changes made since the DataSet was loaded or this method was called.
BeginInit	Begins the initialization of the DataSet. The initialization occurs at run time.
Clear	Clears data.
Clone	Copies the structure of the DataSet, including all DataTable schemas, relations, and constraints. Does not copy any data.
Copy	Copies both structure and data.
CreateDataReader()	Returns a DataTableReader with one result set per DataTable, in the same sequence as the tables appear in the Tables collection.
CreateDataReader(DataTable[])	Returns a DataTableReader with one result set per DataTable.

EndInit	Ends the initialization of the data set.
Equals(Object)	Determines whether the specified Object is equal to the current Object.
Finalize	Free resources and perform other cleanups.
GetChanges	Returns a copy of the DataSet with all changes made since it was loaded or the AcceptChanges method was called.
GetChanges(DataRowState)	Gets a copy of DataSet with all changes made since it was loaded or the AcceptChanges method was called, filtered by DataRowState.
GetDataSetSchema	Gets a copy of XmlSchemaSet for the DataSet.
GetObjectData	Populates a serialization information object with the data needed to serialize the DataSet.
GetType	Gets the type of the current instance.
GetXML	Returns the XML representation of the data.
GetXMLSchema	Returns the XSD schema for the XML representation of the data.
HasChanges()	Gets a value indicating whether the DataSet has changes, including new, deleted, or modified rows.
HasChanges(DataRowState)	Gets a value indicating whether the DataSet has changes, including new, deleted, or modified rows, filtered by DataRowState.

IsBinarySerialized	Inspects the format of the serialized representation of the DataSet.
Load(IDataReader, LoadOption, DataTable[])	Fills a DataSet with values from a data source using the supplied IDataReader, using an array of DataTable instances to supply the schema and namespace information.
Load(IDataReader, LoadOption, String[])	Fills a DataSet with values from a data source using the supplied IDataReader, using an array of strings to supply the names for the tables within the DataSet.
Merge()	Merges the data with data from another DataSet. This method has different overloaded forms.
ReadXML()	Reads an XML schema and data into the DataSet. This method has different overloaded forms.
ReadXMLSchema(0)	Reads an XML schema into the DataSet. This method has different overloaded forms.
RejectChanges	Rolls back all changes made since the last call to AcceptChanges.
WriteXML()	Writes an XML schema and data from the DataSet. This method has different overloaded forms.
WriteXMLSchema()	Writes the structure of the DataSet as an XML schema. This method has different overloaded forms.

The DataTable Class

The DataTable class represents the tables in the database. It has the following important properties; most of these properties are read only properties except the PrimaryKey property:

Properties	Description
ChildRelations	Returns the collection of child relationship.
Columns	Returns the Columns collection.
Constraints	Returns the Constraints collection.
DataSet	Returns the parent DataSet.
DefaultView	Returns a view of the table.
ParentRelations	Returns the ParentRelations collection.
PrimaryKey	Gets or sets an array of columns as the primary key for the table.
Rows	Returns the Rows collection.

The following table shows some important methods of the DataTable class:

Methods	Description
AcceptChanges	Commits all changes since the last AcceptChanges.
Clear	Clears all data from the table.
GetChanges	Returns a copy of the DataTable with all changes made since the AcceptChanges method was called.

GetErrors	Returns an array of rows with errors.
ImportRows	Copies a new row into the table.
LoadDataRow	Finds and updates a specific row, or creates a new one, if not found any.
Merge	Merges the table with another DataTable.
NewRow	Creates a new DataRow.
RejectChanges	Rolls back all changes made since the last call to AcceptChanges.
Reset	Resets the table to its original state.
Select	Returns an array of DataRow objects.

The DataRow Class

The DataRow object represents a row in a table. It has the following important properties:

Properties	**Description**
HasErrors	Indicates if there are any errors.
Items	Gets or sets the data stored in a specific column.
ItemArrays	Gets or sets all the values for the row.
Table	Returns the parent table.

The following table shows some important methods of the DataRow class:

Methods	Description
AcceptChanges	Accepts all changes made since this method was called.
BeginEdit	Begins edit operation.
CancelEdit	Cancels edit operation.
Delete	Deletes the DataRow.
EndEdit	Ends the edit operation.
GetChildRows	Gets the child rows of this row.
GetParentRow	Gets the parent row.
GetParentRows	Gets parent rows of DataRow object.
RejectChanges	Rolls back all changes made since the last call to AcceptChanges.

The DataAdapter Object

The DataAdapter object acts as a mediator between the DataSet object and the database. This helps the Dataset to contain data from multiple databases or other data source.

The DataReader Object

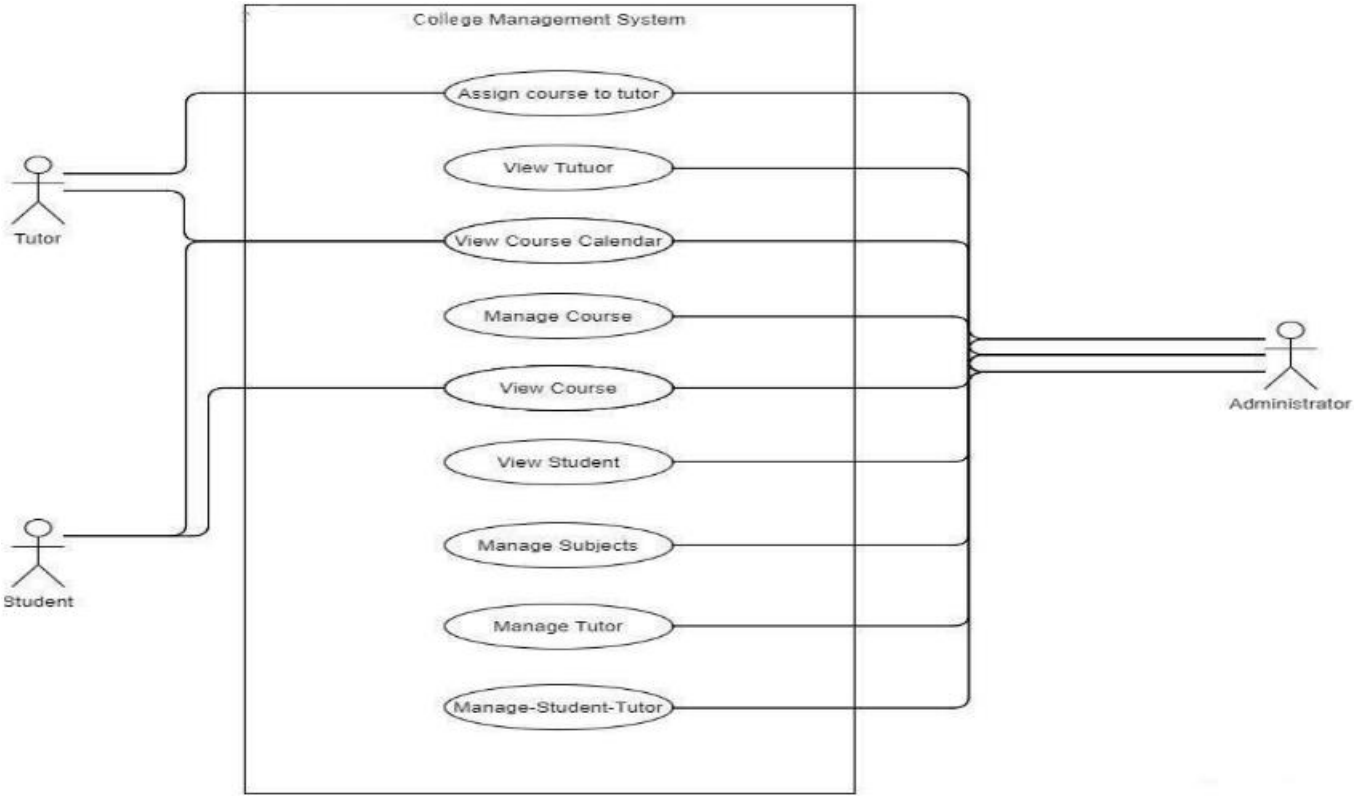
The DataReader object is an alternative to the DataSet and DataAdapter combination. This object provides a connection oriented access to the data records in the database. These objects are suitable for read-only access, such as populating a list and then breaking the connection.

DbCommand and DbConnection Objects

The DbConnection object represents a connection to the data source. The connection could be shared among different command objects.

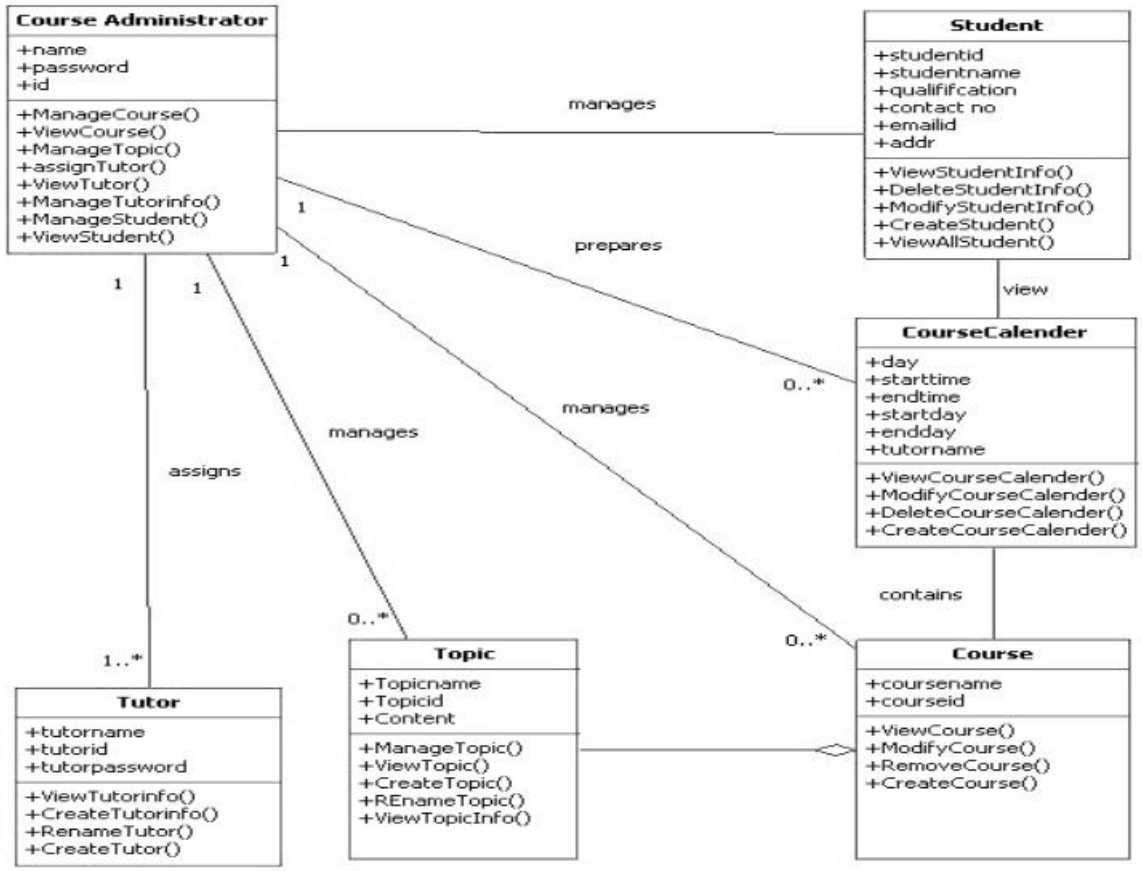
The DbCommand object represents the command or a stored procedure sent to the database from retrieving or manipulating data.

3. Pre-lab Task:

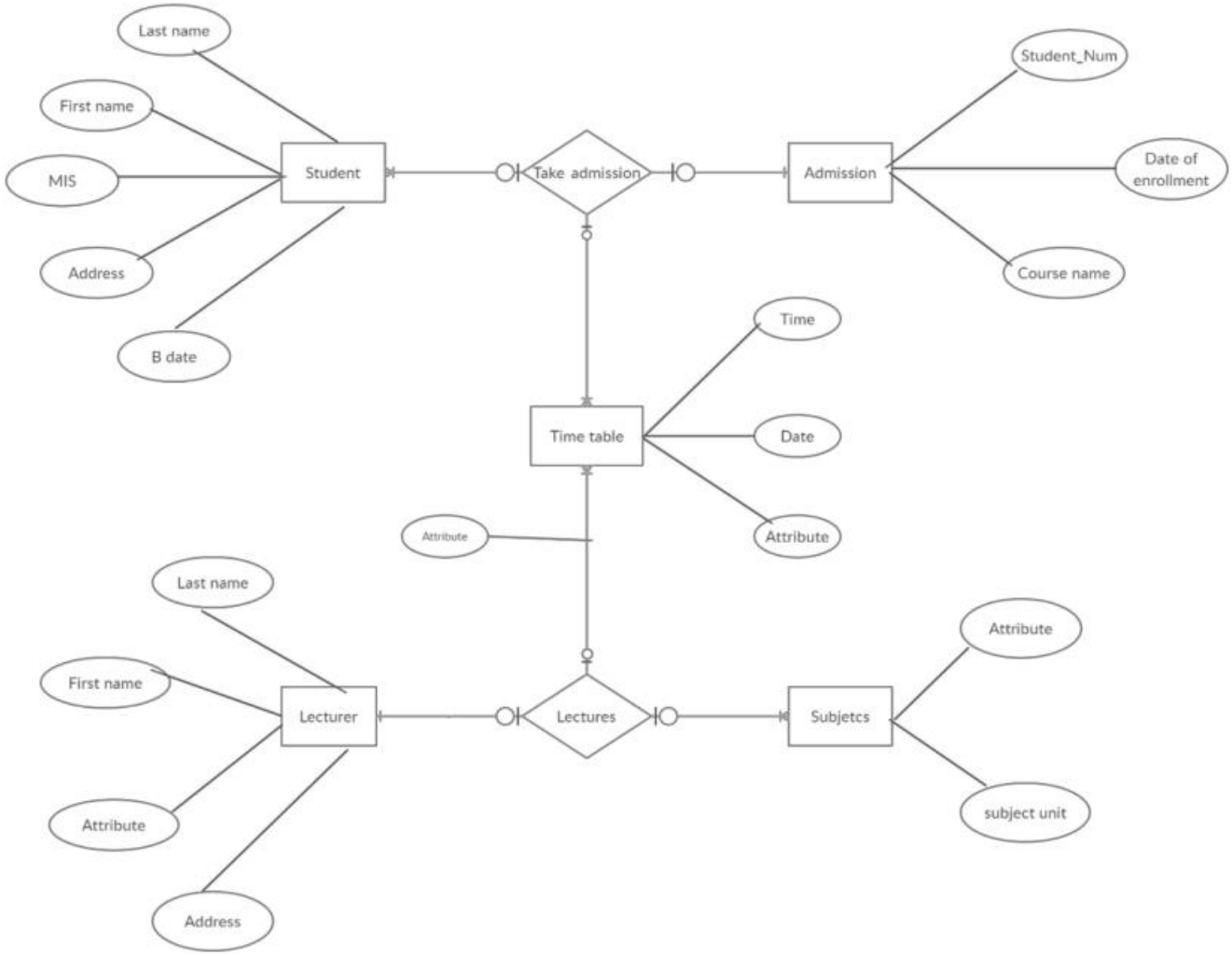


Use case diagram for College management system

1. Class Diagram for College management system



2. E-R diagram for College management system



3. Post Lab Task

Login Form

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="Login.aspx.cs" Inherits="WebApplication2.WebForm2" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">

    <style type="text/css">

        .style1

        {

            width: 74%;

            border-collapse: collapse;
            border: 1px solid #808000;
            margin-left: 0px;
            background-color: #C0C0C0;

        }

        .style2

        {
```

```
            width: 174px;
            color: #800000;

        font-family: "Times New Roman", Times, serif;
        font-size: x-large;

    }

        .style3

        {

            width: 7px;

            text-align: center;

        }

        .style4

        {

            width: 375px;

        }

        .style5

        {

            width: 174px;
            color: #CC0000;
            font-size: small;

        }

        .style6

        {

            width: 174px;
            color: #FF0000;
            height: 28px;

        }

        .style7

        {

            width: 7px;

            text-align: center;
            height: 28px;

        }

        .style8

        {

            width: 375px;
            height: 28px;
        }

    .style10

    {

        width: 174px;
        color: #CC0000;
```

```
            font-size: large;

                text-align: right;

            }

            .style11

            {

                width: 174px;
                color: #FF0000;

                font-family: "Times New Roman", Times, serif;
                font-size: x-large;

                text-align: right;

            }

        </style>

    </asp:Content>

    <asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

        <table align="center" class="style1">

            <tr>

                <td class="style5">
                     </td>

                <td class="style3">
                     </td>

                <td class="style4">

                    <asp:Label ID="lbl" runat="server"

                        style="color: #FF0000; font-size: x-large; font-weight: 700"
    Text="Label"></asp:Label>

                </td>

            </tr>

            <tr>

                <td class="style10">

                    <strong>User Name</strong></td>

                <td class="style3">

                    :</td>

                <td class="style4">

                    <asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>

                    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
                        ErrorMessage="Enter User Name"

    ControlToValidate="txtUserName"></asp:RequiredFieldValidator>

                </td>

            </tr>

            <tr>

                <td class="style11">

                    <strong>Password</strong></td>
```

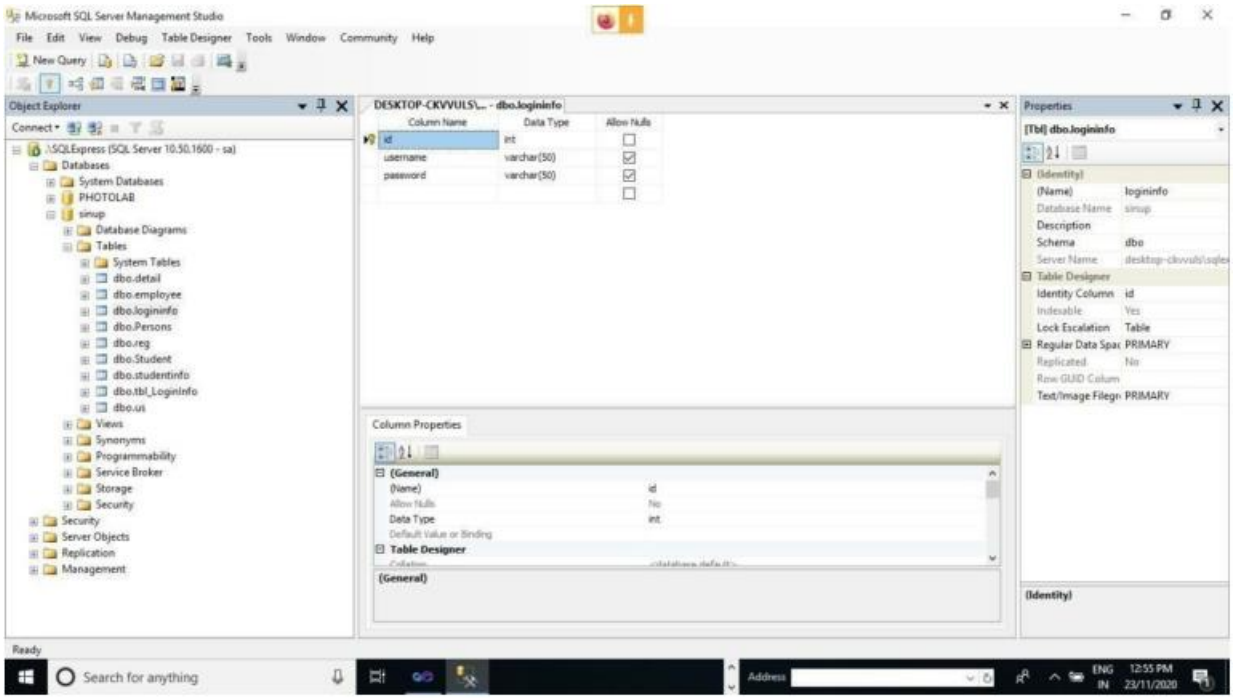
```
                <td class="style3">

                    :</td>

                <td class="style4">

                    <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>

                    <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
                        ErrorMessage="Enter Passwrd"

    ControlToValidate="txtUserName"></asp:RequiredFieldValidator>

                </td>

            </tr>

            <tr>

                <td class="style6">

                </td>

                <td class="style7">

                </td>

                <td class="style8">

                    <asp:Button ID="btnSubmit" runat="server" Text="Submit"
                        onclick="btnSubmit_Click" />

                    <asp:ValidationSummary ID="ValidationSummary1" runat="server"
                        ShowMessageBox="True" />

                </td>

            </tr>

        </table>

    </asp:Content>
```

Login Form Asp.net Code:

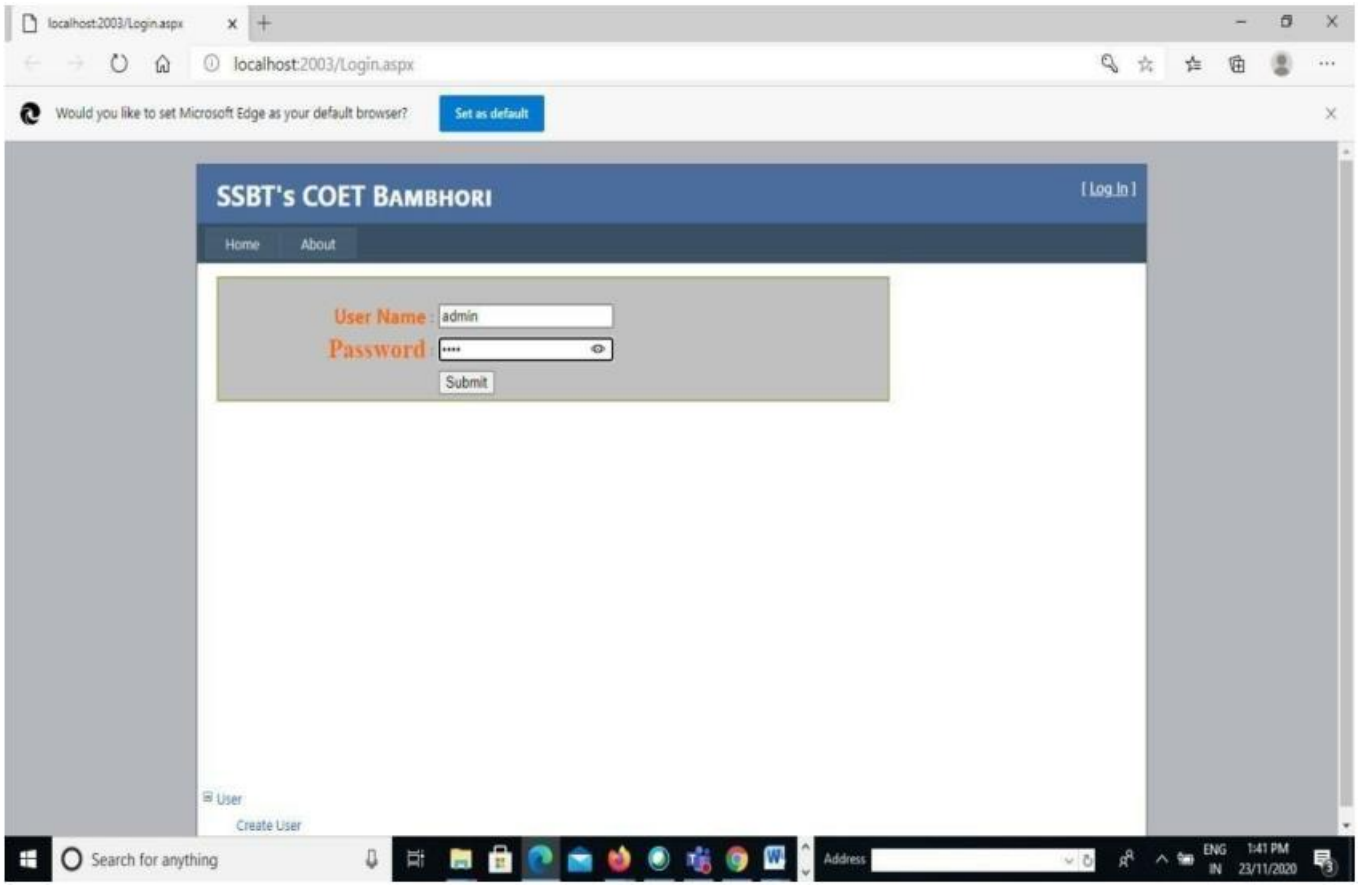
```
using System;

using System.Collections.Generic;
using System.Linq;

using System.Web;
using System.Web.UI;

using System.Web.UI.WebControls;
using System.Data;

using System.Data.SqlClient;


namespace WebApplication2

{

    public partial class WebForm2 : System.Web.UI.Page

    {

        SqlConnection con = new SqlConnection("Data Source=.\\SQLExpress;Initial
Catalog=sinup;User ID=sa;Password=ssbt");

        protected void Page_Load(object sender, EventArgs e)

        {

            lbl.Visible = false;

        }

        public void show()

        {

            try

            {

                SqlCommand sqlcmd = new SqlCommand("insert into logininfo values('" +
txtUserName.Text + "','" + txtPassword.Text + "')", con);

                con.Open();
                sqlcmd.ExecuteNonQuery();
                con.Close();

            }

            catch (Exception ex)

            {

                lbl.Text = ex.Message;

            }

            finally

            {

                con.Close();
                con.Dispose();

            }


        }
```

```
        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            show();
        }
    }
}
```

Database



Output


localhost:2003/Login.aspx
Would you like to set Microsoft Edge as your default browser?
Set as default
SSBT's COET BAMBHORI
[Log In]
Home
About
User Name :
admin
Password
Submit
User
Create User
Search for anything
Address
ENG
IN
1:41 PM
23/11/2020

Information Form Design

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="info.aspx.cs" Inherits="WebApplication2.WebForm4" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">

    <style type="text/css">

        .style1

        {

            width: 100%;

        }

        .style2

        {

            width: 41px;

            text-align: center;

        }

        .style3

        {

            width: 232px;

        }

        .style4

        {

            font-family: "Times New Roman";
            font-weight: bold;

            color: #00FF00;
            font-size: x-large;

            background-color: #CC0000;

        }

        .style5

        {

            background-color: #CC3300;

        }

        .style6

        {

            width: 41px;

            text-align: center;
            color: #000000;
            font-size: x-large;
        }
            background-color: #669900;
    </style>
```

```
    </asp:Content>
    <asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
        <table class="style1">
            <tr>
                <td class="style4" colspan="3" width="30">
                    <asp:Label ID="lbl" runat="server"></asp:Label>
                </td>
            </tr>
            <tr>
                <td class="style4" width="30">
                    Roll No</td>
                <td class="style6">
                    :</td>
                <td class="style5">
                    <asp:TextBox ID="txtRoll" runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td class="style4" width="30">
                    Student Name</td>

                <td class="style6">
                    :</td>
                <td class="style5">
                    <asp:TextBox ID="txtStudentName" runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td class="style4" width="30">
                    Father Name</td>
                <td class="style6">
                    :</td>
                <td class="style5">
                    <asp:TextBox ID="txtFatherName" runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td class="style4" width="30">
                    City</td>
                <td class="style6">
```

```
                    :</td>
                <td class="style5">
                    <asp:TextBox ID="txtCity" runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td class="style4" width="30">
                     </td>
                <td class="style6">
                     </td>
                <td class="style5">
                    <asp:Button ID="btnSave" runat="server" onclick="btnSave_Click"
                        style="background-color: #FF9900" Text="Save" />
    
                </td>
            </tr>
            <tr>
                     </td>
                <td class="style3" width="30">
                     </td>
                <td class="style2">
                     </td>
                                                                                    <td>
            </tr>
        </table>
    </asp:Content>
```

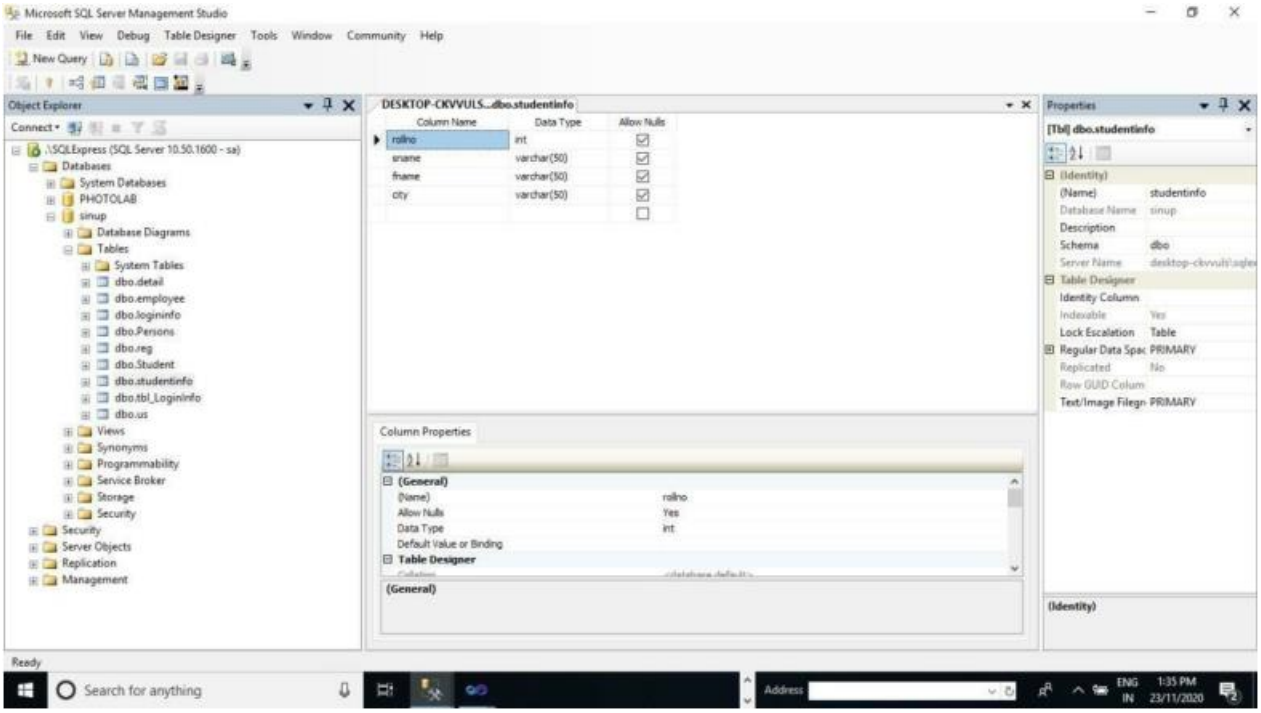
Information Form Asp.net Code

```
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Web;
    using System.Web.UI;
    using System.Web.UI.WebControls;
    using System.Data;
    using System.Data.SqlClient;

    namespace WebApplication2
    {
        public partial class WebForm4 : System.Web.UI.Page
        {
```

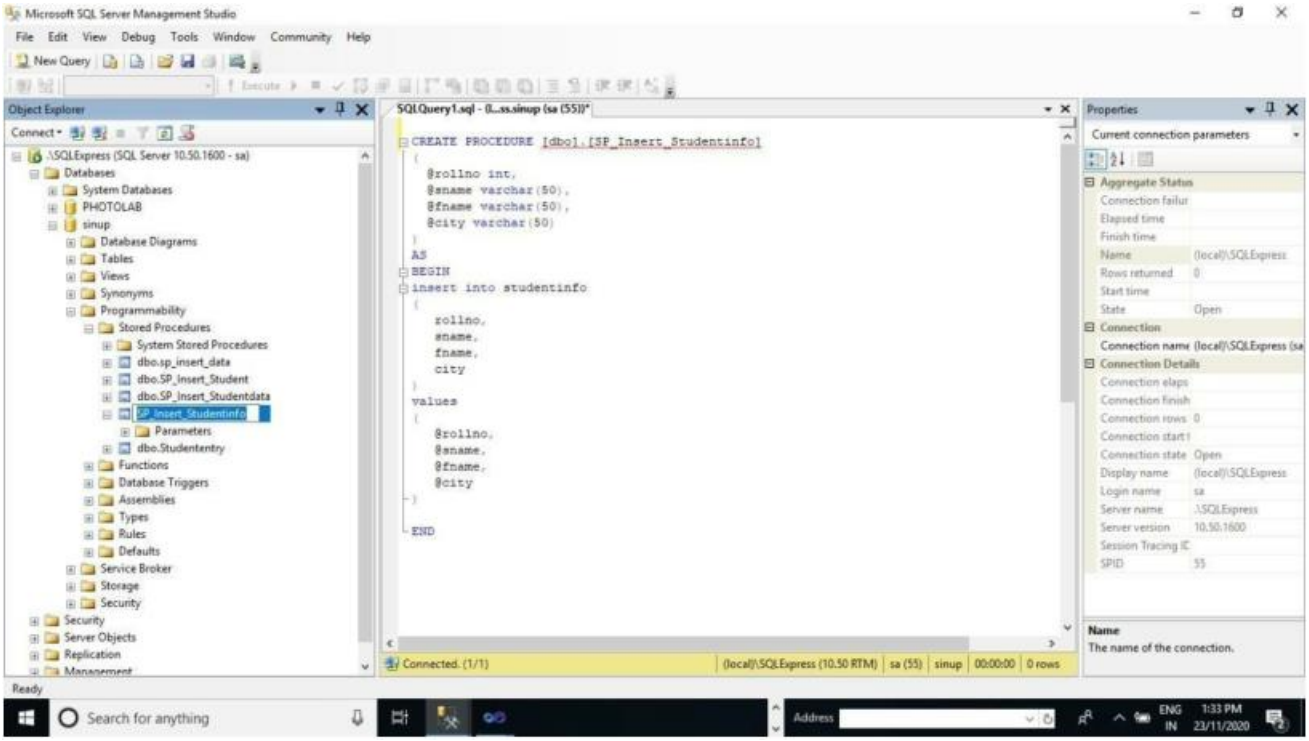
```
        protected void Page_Load(object sender, EventArgs e)

        {

            lbl.Visible = false;


        }


        protected void btnSave_Click(object sender, System.EventArgs e)

        {

            SqlConnection con = new SqlConnection("Data Source=.\\SQLExpress;Initial
Catalog=sinup;User ID=sa;Password=ssbt");

            SqlCommand sqlcom = new SqlCommand("SP_Insert_Studentinfo", con);
            con.Open();

            sqlcom.CommandType = CommandType.StoredProcedure;
            sqlcom.Parameters.AddWithValue("@rollno", txtRoll.Text.ToString());
            sqlcom.Parameters.AddWithValue("@sname", txtStudentName.Text.ToString());
            sqlcom.Parameters.AddWithValue("@fname",txtFatherName.Text.ToString());
            sqlcom.Parameters.AddWithValue("@city",txtCity.Text.ToString());
            sqlcom.ExecuteNonQuery();

            con.Close();

            lbl.Text = "Record Inserted Successfully";
            lbl.Visible = true;

            clearcontrol();


        }

        public void clearcontrol()


        {

            txtCity.Text = "";
            txtFatherName.Text = "";
            txtRoll.Text = "";
            txtStudentName.Text = "";

        }


    }

}
```

Information Form Database:

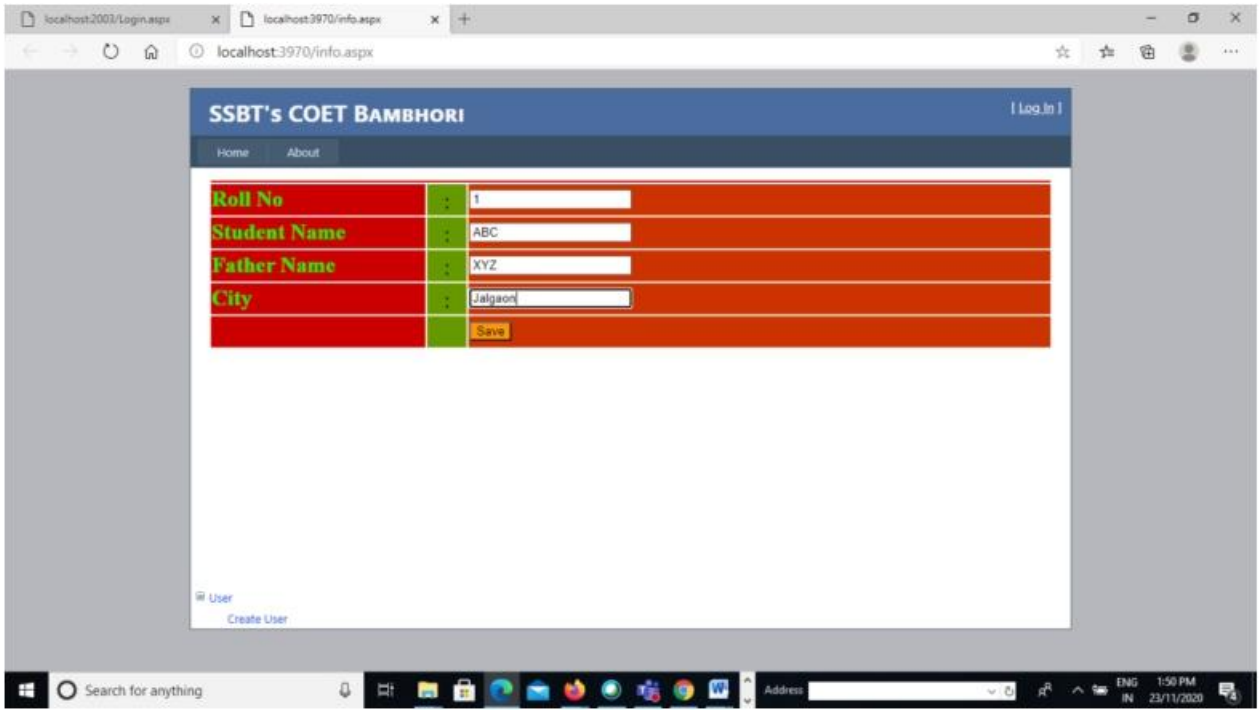
Create Table



Stored Procedure



Output



Report Design

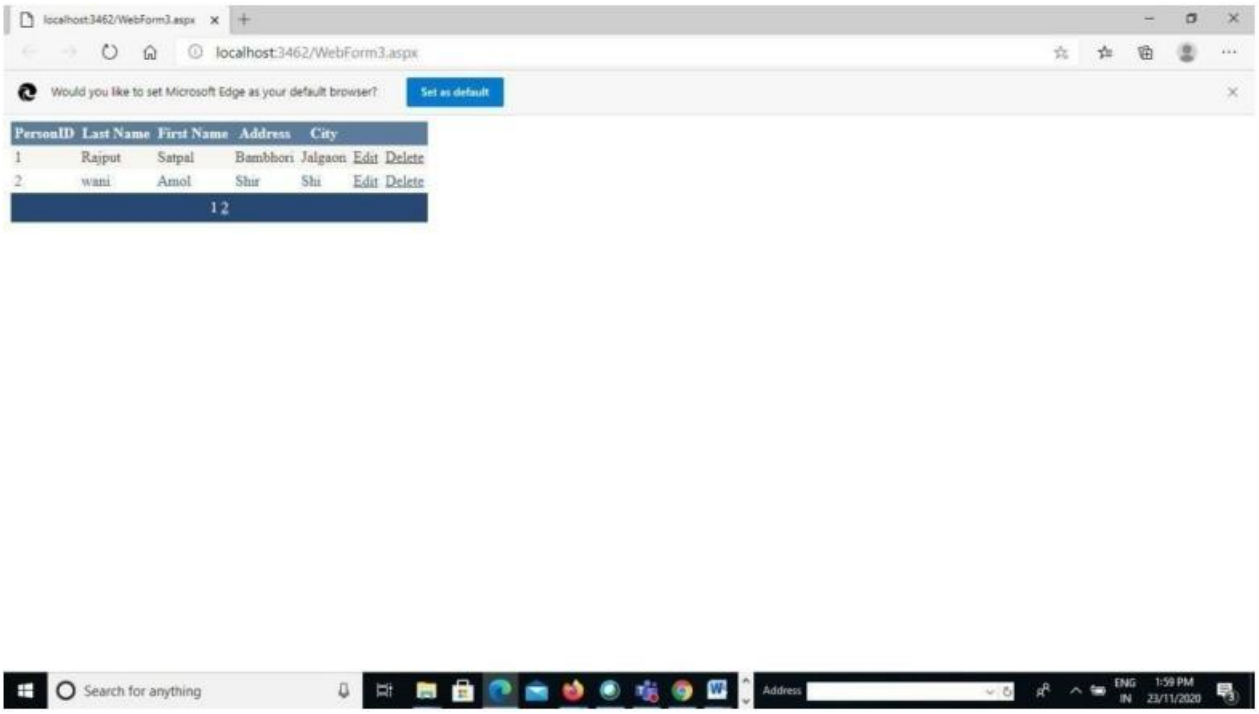
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm3.aspx.cs"
Inherits="WebApplication1.WebForm3" %>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">


<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title></title>

</head>

<body>

    <form id="form1" runat="server">

    <div>


        <asp:GridView ID="gvData" runat="server" AllowPaging="True"
            AutoGenerateColumns="False" CellPadding="4" ForeColor="#333333"
            GridLines="None" onpageindexchanging="gvData_PageIndexChanging" PageSize="2">

            <AlternatingRowStyle BackColor="White" ForeColor="#284775" />

            <Columns>

                <asp:BoundField DataField="PersonID" HeaderText="PersonID" />

                <asp:BoundField DataField="LastName" HeaderText="Last Name" />

                <asp:BoundField DataField="FirstName" HeaderText="First Name" />

                <asp:BoundField DataField="Address" HeaderText="Address" />

                <asp:BoundField DataField="City" HeaderText="City" />
```

```
                <asp:CommandField ShowEditButton="true" />
                <asp:CommandField ShowDeleteButton="true" />
            </Columns>
            <EditRowStyle BackColor="#999999" />
            <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
            <HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
            <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
            <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
            <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />
            <SortedAscendingCellStyle BackColor="#E9E7E2" />
            <SortedAscendingHeaderStyle BackColor="#506C8C" />
            <SortedDescendingCellStyle BackColor="#FFFDF8" />
            <SortedDescendingHeaderStyle BackColor="#6F8DAE" />
        </asp:GridView>

    </div>
    </form>
</body>
</html>
```

Report Asp.net Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
namespace WebApplication1
{
    public partial class WebForm3 : System.Web.UI.Page
    {
```

```
        SqlConnection con = new SqlConnection("Data
Source=.\\SQLExpress;Initial Catalog=sinup;User ID=sa;Password=ssbt");

        protected void Page_Load(object sender, EventArgs e)

        {

            con.Open();

            show();

        }

        private void show()

        {

            SqlCommand cmd = new SqlCommand("select * from Persons",
            con); SqlDataAdapter da = new SqlDataAdapter(cmd);

            DataSet ds = new
            DataSet();
            da.Fill(ds);
            gvData.DataSource =
            ds;
            gvData.DataBind();


        }


        protected void gvData_PageIndexChanging(object sender, GridViewPageEventArgs e)

        {

            gvData.PageIndex =
            e.NewPageIndex; show();


        }


    }


}


'Output
```



Outcomes:

1. Creates ASP.net Web Forms
2. Created ADO.net programs that use various library functions, and that manipulate database

Reference:

1) https://learn.microsoft.com/en-us/aspnet/web-forms/

2) https://learn.microsoft.com/en-us/aspnet/web-forms/videos/

3) Kudvenkat youtube channel

Experiment No. 4

Aim: Implementation of web Application using R**.**

1. Objective: Development of web Application using R.

2. Background:

R: It is a programming language and software environment for statistical analysis, graphics representation and reporting. R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. The core of R is an interpreted computer language which allows branching and looping as well as modular programming using functions. R allows integration with the procedures written in the C, C++, .Net, Python or FORTRAN languages for efficiency. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems like Linux, Windows and Mac. R is free software distributed under a GNU-style copy left, and an official part of the GNU project called GNU S. As stated earlier, R is a programming language and software environment for statistical analysis, graphics representation and reporting.

The following are the important features of R:

- R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
- It has an effective data handling and storage facility,
- It provides a suite of operators for calculations on arrays, lists, vectors and matrices.
- It provides a large, coherent and integrated collection of tools for data analysis.
- It provides graphical facilities for data analysis and display directly on the computer.

R is world's most widely used statistics programming language. It's the first choice of data scientists and supported by a vibrant and talented community of contributors. R is taught in universities and deployed in mission critical business applications.

Why use R Programming?

There are several tools available in the market to perform data analysis. Learning new languages is time taken. The data scientist can use two excellent tools, i.e., R and Python. We may not have time to learn them both at the time when we get started to learn data science. Learning statistical modeling and algorithm is more important than to learn a programming language. A programming language is used to compute and communicate our discovery.

The important task in data science is the way we deal with the data: clean, feature engineering, feature selection, and import. It should be our primary focus. Data scientist job is to understand the data, manipulate it, and expose the best approach. For machine learning, the best algorithms can be implemented with R. Keras and TensorFlow allow us to create high-end machine learning techniques. R has a package to perform Xgboost. Xgboost is one of the best algorithms for Kaggle competition.

R communicate with the other languages and possibly calls Python, Java, C++. The big data world is also accessible to R. We can connect R with different databases like **Spark** or **Hadoop**. In brief, R is a great tool to investigate and explore the data. The elaborate analysis such as clustering, correlation, and data reduction are done with R. There are several-applications available in real-time. Some of the popular applications are: Facebook, Google, Twitter, HRDAG, Sunlight Foundation, RealClimate, NDAA,etc.

3. Pre-lab Task:

Installation of R-base for windows

1. Goto link http://cran.r-project.org
2. Select “Download for Windows” & Click on “Base”
3. Select“Download R 3.3.2 for Windows(62 megabytes, 32/64 bit)”
4. Click on “Save file”
5. After downloading double click on “R 3.3.2-win.exe” file
6. Click 'next' ..... 'Finish'.

Installation of R-studio for windows

1. Go to RStudio website (https://www.rstudio.com/products/rstudio/download/),
2. Select Platform “Rstudio 1.0.136-windows vista/7/8/10”
3. Double click on “Rstudio 1.0.136-windows vista/7/8/10”
4. Click next to continue when the install wizard opens.
5. Click next to accept the default install location.
6. Click Install to accept the default start menu folder and install RStudio! Click Finish to close the wizard.

Installation of R-base for Ubuntu

1. sudo apt-get install r-base
2. if require- sudo apt-get update

Through Ubuntu Software Center:

1. Open Ubuntu Software Center.
2. Search for r-base & click Install.
3. Then run R by executing R in the Terminal.

Installation of R-studio for Ubuntu

1. Go to RStudio website choose and download the Rstudio Desktop version for your system (www.rstudio.com/products/rstudio/download/)
2. Open this file in Ubuntu Software Center
3. Click install and you're done

Installation Open source R package (i.e. R-shiny)

Run the following command from an R console:

install.packages("shiny")

Data Types and R-Objects

In R programming language the variables are not declared as some data type. The variables are assigned with R-Objects. The data type of the R-object becomes the data type of the variable. There are many types of R-objects. Following are some of the Data types and R-Objects.

Table: Data types and R-Objects

	Data types	R- Objects
1.	Logical	Vectors
2.	Numeric	Lists
3.	Integer	Matrices
4.	Character	Arrays
5.	Complex	Data Frames

4. In-lab Task:

Create a simple app that displays text within the title panel, sidebar panel and main panel, where the sidebar panel is located on the right

Step 1: Install package & build framework

```
>install.packages("shiny")
>library(shiny)
>ui ←fluidPage()
>server ←function (input, output) {}
>shinyApp (ui =ui , server = server)
```

Step 2: Building the UI framework

```
>install.packages("shiny")
>library(shiny)
>ui ←fluidPage(
        titlePanel(title = "First app….."),
        sidebarLayout(
        sidebarPanel ("Sidebar panel, ….."),
        mainPanel("Main panel, ….."))
        ))
>server ←function (input, output) {}
>shinyApp (ui =ui , server = server)
```

Step 3: Adjusting the UI framework

```
>install.packages("shiny")
>library(shiny)
>ui ←fluidPage(
        titlePanel(title = "First app….."),
        sidebarLayout( position = "right",
        sidebarPanel ("Sidebar panel, ….."),
        mainPanel("Main panel, ….."))
        ))
>server ←function (input, output) {}
>shinyApp (ui =ui , server = server)
```

5. Post-lab Task:

Outcomes:

Student should able

1. To execute R script.
2. To implement web app using R

Web Resources:

1. https://www.dezyre.com/projects/data-science-projects/data-science-projects-in-r
2. https://data-flair.training/blogs/data-science-projects__code/
3. https://machinelearningmastery.com/machine-learning-in-r-step-by-step/
4. Krish Naik Hindi Youtube channel https://www.youtube.com/watch?v=XxM84pBKWog&list=PLTDARY42LDV6tX977fE3q6Sk_6mA7O4o5