



# CMR TECHNICAL CAMPUS

## UGC(AUTONOMOUS)



# PREDICTING EMPLOYEES UNDER STRESS FOR PRE-EMPTIVE REMEDIATION USING MACHINE LEARNING ALGORITHM

BATCH NO:02

GROUP MEMBERS :

RAJA SULAKSHANA - 207R1A0585

MOHANA LAHARI-207R1A0599

UNDER THE GUIDANCE OF:

DR. D.T.V. DHARMAJEE RAO

(PROFESSOR AND DEAN ACADEMICS)

# ABSTRACT

---

- The COVID-19 pandemic forced businesses to adopt unconventional work patterns, including remote work and reduced on-site staffing.
- Employees have had to adapt to these changes, leading to psychological stress and fatigue.
- Data visualization and Naive Bayes based machine learning predict employee stress levels effectively. Key factors for determining stress are working hours, workload, age, and role ambiguity.
- It is concluded that increasing working hours, role ambiguity, and workload would have a detrimental impact on various aspects of employee performance.



# EXISTING SYSTEM

---

- Workplace stress is a significant concern for employees, including human resource managers.
- Data from a diverse sample of employees from various organizations and industries indicates that positive factors like efficacy, optimism, and resilience play a crucial role in understanding stress adaptation.
- Numerous studies and experiments have been conducted in recent years, often in economically and socially advancing countries.
- Stress is increasingly recognized as one of the most common occupational disorders, and previously by using KNN did not achieved good accuracy .

# **DISADVANTAGES**

- The system is not implemented PCA (Principal Component Analysis)
- The system is not implemented XGB Classifier (XS Boost).

# PROPOSED SYSTEM

---

- The proposed system is intended for the commercial sector and is implemented using XGB classifier and PCA(Principal component Analysis).
- By understanding the stress experienced by employees, we may develop strategies or solutions to lessen stress and improve the working environment.
- The system takes into account a variety of factors, including gender, age, financial concerns, family concerns, working hours, learning methods, health concerns, issues with colleagues, pressure, regularity, and interaction, among others.
- Based on the stress levels, the system also makes recommendations to the working staff.
- we have used Naive Bayes to achieve better results and accuracy.

# ADVANTAGES

---

- The goal of the system is to test and train the large number of datasets with high accuracy.
- The proposed system developed a Machine Learning Algorithms to test and train the datasets.

# Hardware and Software Requirements

---

## Software:

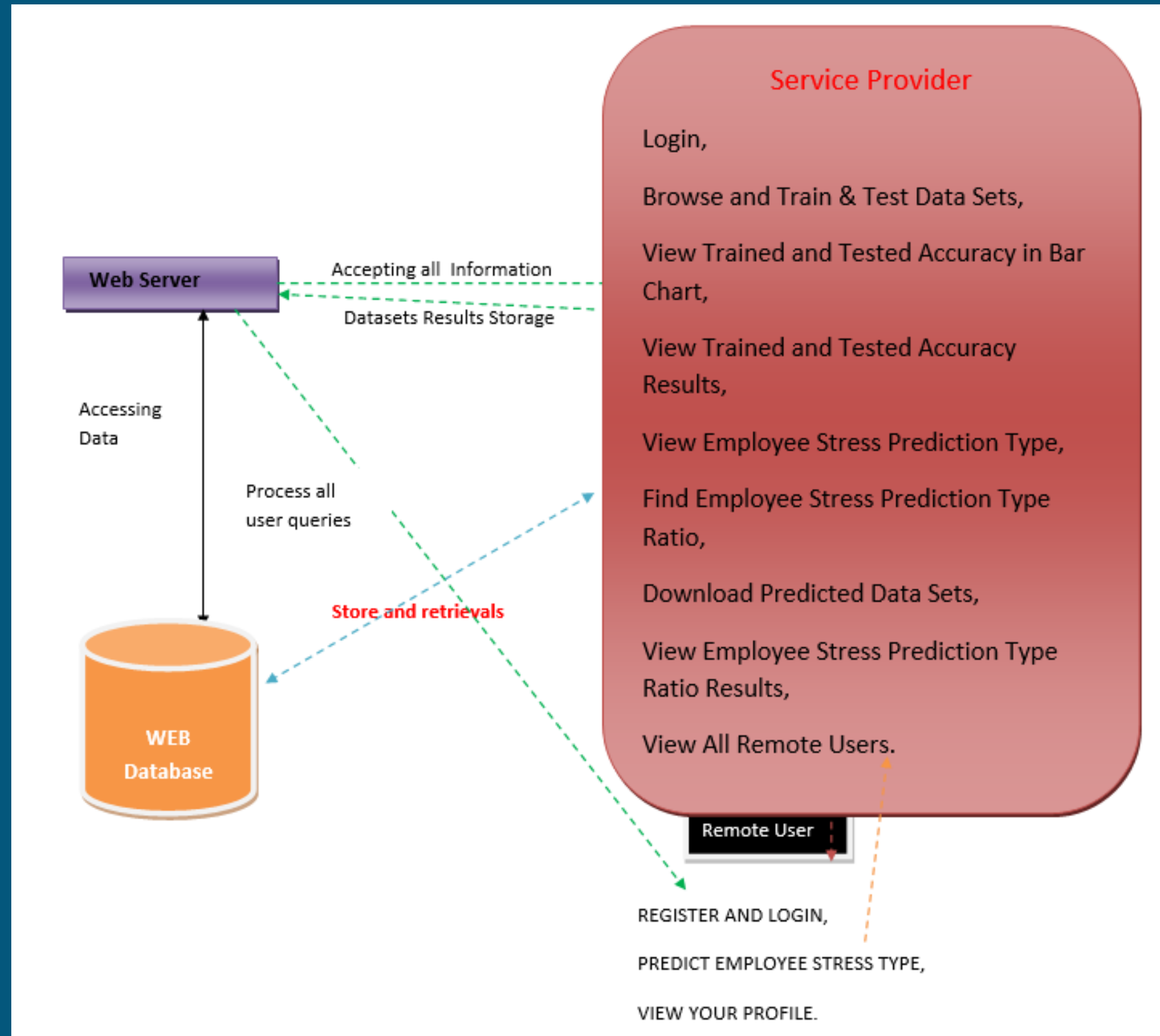
- Operating system : windows 11
- CODING LANGUAGE : Python
- Front-End : Html,CSS,Javascript
- Back-End : Django-ORM
- Data Base : MySQL(WAMP Server)

## Hardware:

- Processor : 11th Gen Intel core i5
- RAM : 8 GB

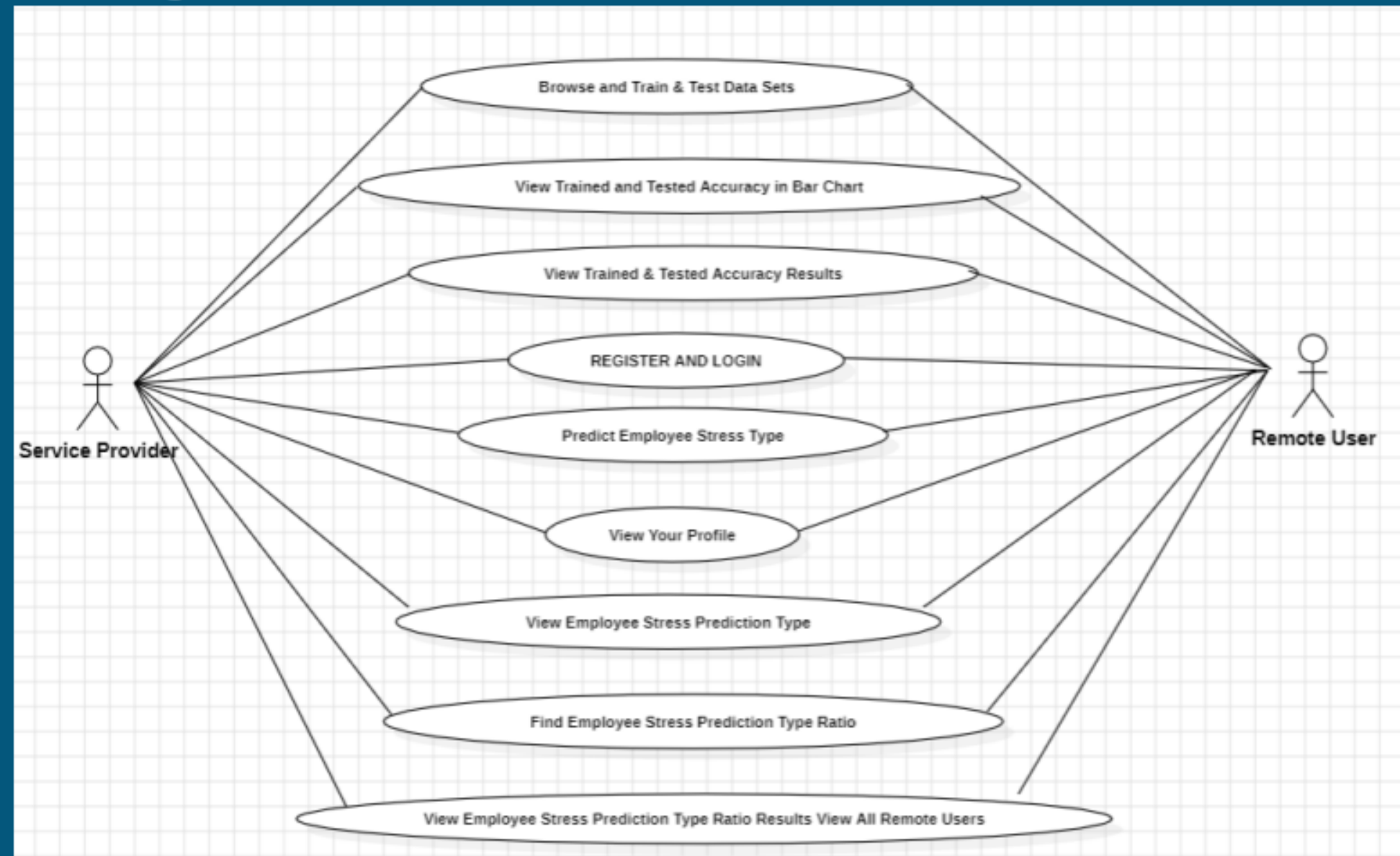
Any System with above or higher Configuration is Compatible for this Project

# SYSTEM ARCHITECTURE

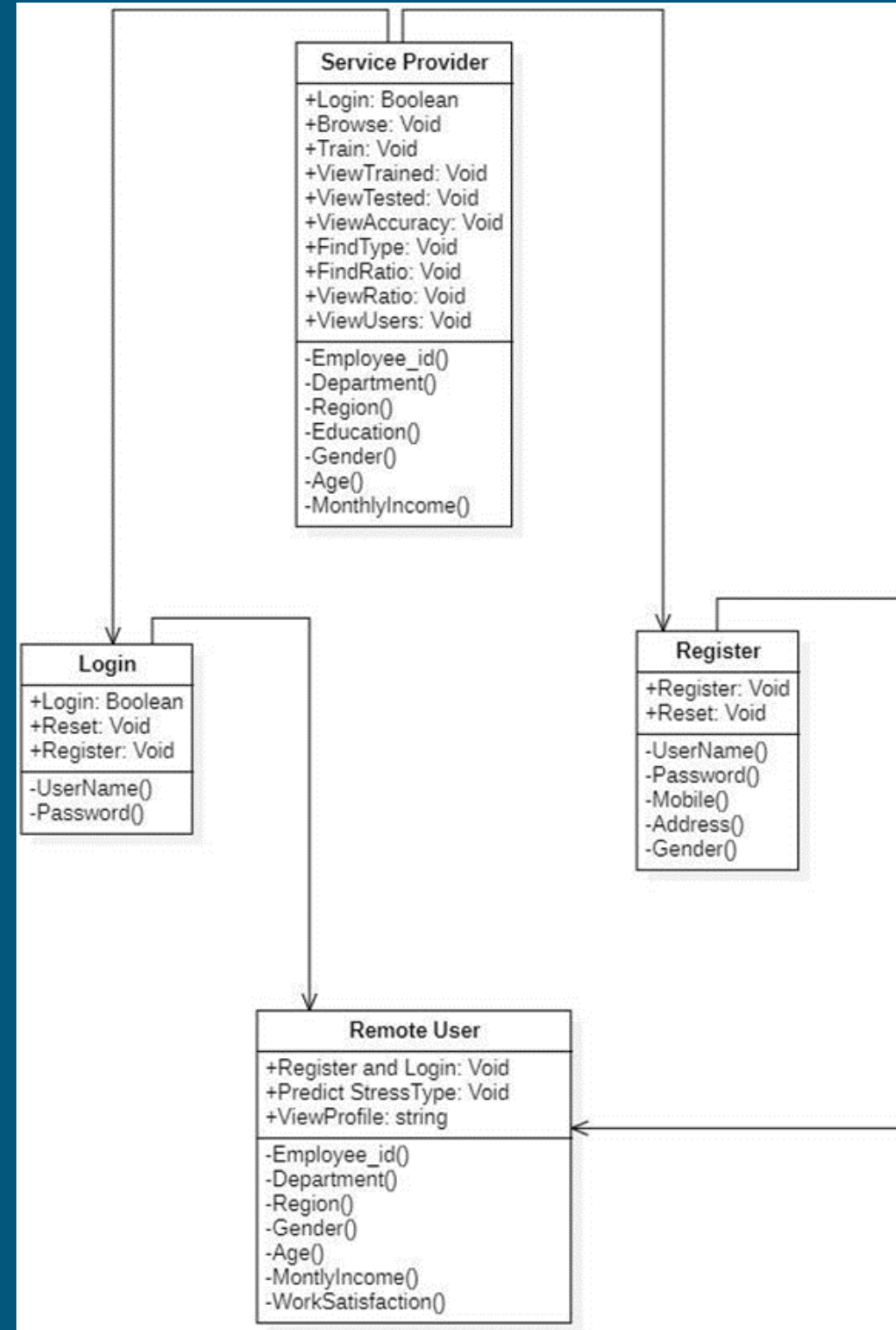




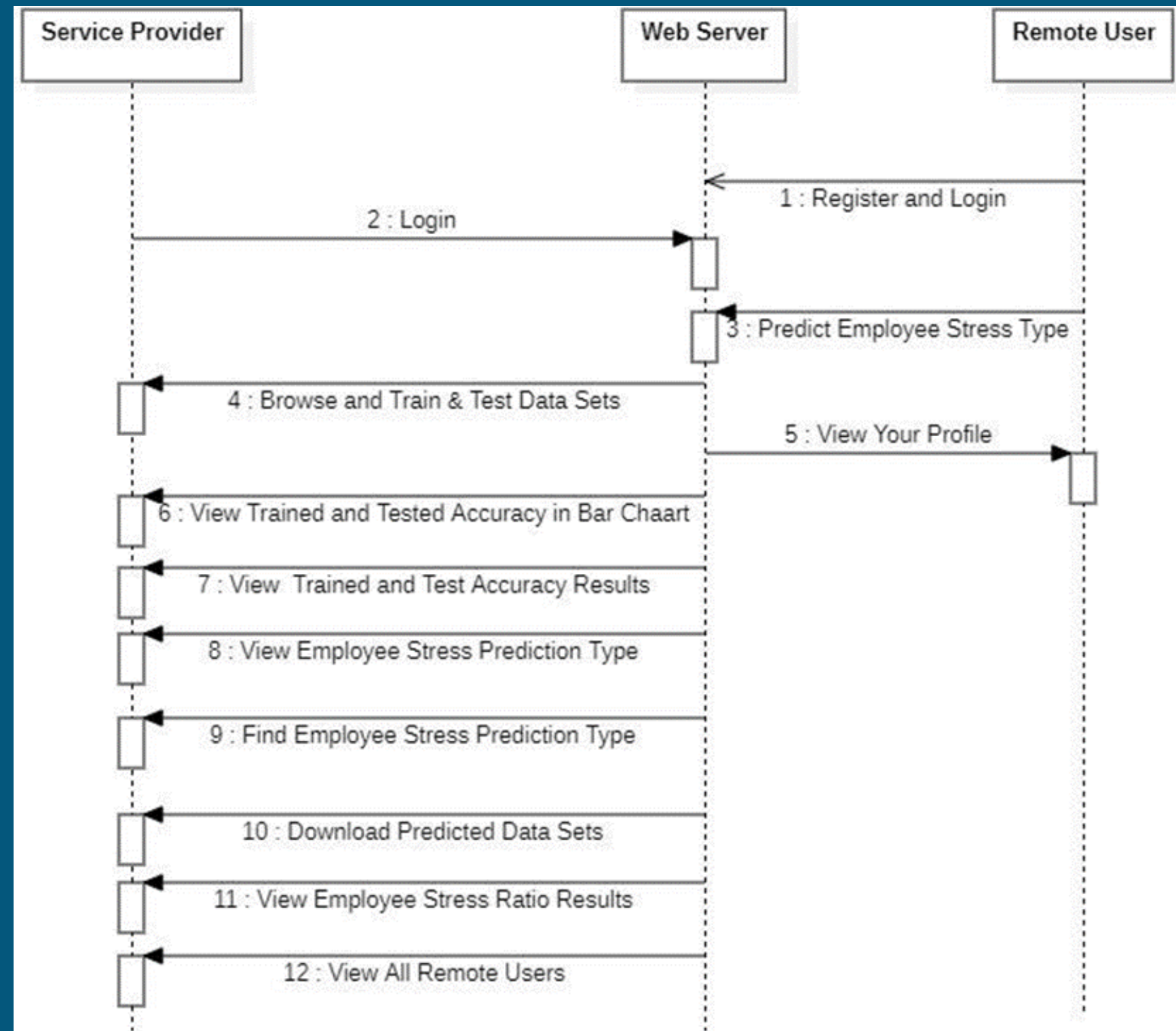
# USECASE DIAGRAM



# CLASS DIAGRAM



# SEQUENTIAL DIAGRAM



# SAMPLE CODE

---

```
#!/usr/bin/env python
"""DJANGO'S COMMAND-LINE UTILITY FOR ADMINISTRATIVE TASKS."""
import os
import sys

def main():
    """RUN ADMINISTRATIVE TASKS."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'predicting_employees_under_stress.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "COULDN'T IMPORT DJANGO. ARE YOU SURE IT'S INSTALLED AND "
            "AVAILABLE ON YOUR PYTHONPATH ENVIRONMENT VARIABLE? DID YOU "
            "FORGET TO ACTIVATE A VIRTUAL ENVIRONMENT?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

```
FROM DJANGO.DB IMPORT MODELS
```

```
# CREATE YOUR MODELS HERE.
```

```
FROM DJANGO.DB.MODELS IMPORT CASCADE
```

```
CLASS CLIENTREGISTER_MODEL(MODELS.MODEL):
```

```
    USERNAME = MODELS.CHARFIELD(MAX_LENGTH=30)
```

```
    EMAIL = MODELS.EMAILFIELD(MAX_LENGTH=30)
```

```
    PASSWORD = MODELS.CHARFIELD(MAX_LENGTH=10)
```

```
    PHONENO = MODELS.CHARFIELD(MAX_LENGTH=10)
```

```
    COUNTRY = MODELS.CHARFIELD(MAX_LENGTH=30)
```

```
    STATE = MODELS.CHARFIELD(MAX_LENGTH=30)
```

```
    CITY = MODELS.CHARFIELD(MAX_LENGTH=30)
```

```
    ADDRESS= MODELS.CHARFIELD(MAX_LENGTH=300)
```

```
    GENDER= MODELS.CHARFIELD(MAX_LENGTH=30)
```

```
CLASS PREDICTING_EMPLOYEE_STRESS(MODELS.MODEL):
```

```
    EMPLOYEE_ID= MODELS.CHARFIELD(MAX_LENGTH=3000)
```

```
    DEPARTMENT= MODELS.CHARFIELD(MAX_LENGTH=3000)
```

```
    REGION= MODELS.CHARFIELD(MAX_LENGTH=3000)
```

```
    EDUCATION= MODELS.CHARFIELD(MAX_LENGTH=3000)
```

```
    GENDER= MODELS.CHARFIELD(MAX_LENGTH=3000)
```

```
RECRUITMENT_CHANNEL= MODELS.CHARFIELD(MAX_LENGTH=3000)
    TRAINING_TIME= MODELS.CHARFIELD(MAX_LENGTH=3000)
        AGE= MODELS.CHARFIELD(MAX_LENGTH=3000)
PRFORMANCE_RATING= MODELS.CHARFIELD(MAX_LENGTH=3000)
YEARS_AT_COMPANY= MODELS.CHARFIELD(MAX_LENGTH=3000)
    WORKING_HOURS= MODELS.CHARFIELD(MAX_LENGTH=3000)
    FLEXIBLE_TIMINGS= MODELS.CHARFIELD(MAX_LENGTH=3000)
    WORKLOAD_LEVEL= MODELS.CHARFIELD(MAX_LENGTH=3000)
    MONTHLY_INCOME= MODELS.CHARFIELD(MAX_LENGTH=3000)
    WORK_SATISFACTION= MODELS.CHARFIELD(MAX_LENGTH=3000)
PERCENT_SALARY_HIKE= MODELS.CHARFIELD(MAX_LENGTH=3000)
COMPANIES_WORKED= MODELS.CHARFIELD(MAX_LENGTH=3000)
    MARITAL_STATUS= MODELS.CHARFIELD(MAX_LENGTH=3000)
    PREDICTION= MODELS.CHARFIELD(MAX_LENGTH=3000)
```

```
FROM DJANGO IMPORT FORMS
```

```
FROM REMOTE_USER.MODELS IMPORT CLIENTREGISTER_MODEL
```

```
    CLASS CLIENTREGISTER_FORM(FORMS.MODELFORM):
PASSWORD = FORMS.CHARFIELD(WIDGET=FORMS.PASSWORDINPUT())
        EMAIL = FORMS.EMAILFIELD(REQUIRED=TRUE)
```

```
        CLASS META:
            MODEL = CLIENTREGISTER_MODEL
FIELDS = ("USERNAME","EMAIL","PASSWORD","PHONENO","COUNTRY","STATE","CITY")
```

# RESULTS

## Predicting Employees under Stress for Pre-emptive Remediation using Machine learning Algorithm

[Browse and Train & Test Data Sets](#)[View Trained and Tested Accuracy in Bar Chart](#)[View Trained and Tested Accuracy Results](#)[View Employee Stress Prediction Type](#)[Find Employee Stress Prediction Type Ratio](#)[Download Predicted Data Sets](#)[View Employee Stress Prediction Type Ratio Results](#)[View All Remote Users](#)[Logout](#)

### View Employee Stress Type Ratio Details

Employee Stress Prediction Type	Ratio
Low Stress	14.285714285714285
More Stress	85.71428571428571







# REFERENCES

---

[1] SHEKHAR PANDEY, SUPRIYA MUTHURAMAN, ABHILASH SHRIVASTAVA. THE INTERNATIONAL SYMPOSIUM ON INTELLIGENT SYSTEMS TECHNOLOGIES AND APPLICATIONS (2018), DOI: 10.1007/978-3-319-68385-0\_10.

[2] RAMACHANDRAN, R; RAJEEV, D.C; KRISHNAN, S.G; SUBATHRA.P. INTERNATIONAL JOURNAL OF APPLIED ENGINEERING RESEARCH (2015), RESEARCH INDIA PUBLICATIONS, VOLUME 10, NUMBER 10, P.25433-25448

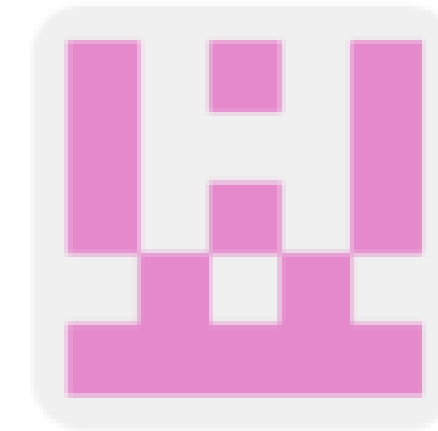
[3] RAMIN ZIBASERESHT: HOW TO RESPOND TO THE ONGOING PANDEMIC OUTBREAK OF THE CORONAVIRUS DISEASE (COVID-19) (WHO- WORLD HEALTH ORGANIZATION) (2020), ISSN 2349- 8870.

[4] CHEN, TIANQI; GUESTRIN, CARLOS; "XG BOOST: A SCALABLE TREE BOOSTING SYSTEM". PROCEEDINGS OF THE 22ND ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, SAN FRANCISCO, USA (2016). ACM. PP. 785-794.

# GITHUB LINK

---

Sulakshana3007/  
**Employees-stress-...**



1

Contributor



0

Issues



0

Stars



0

Forks



## **Sulakshana3007/Employees-stress-prediction**

Contribute to Sulakshana3007/Employees-stress-prediction development by creating an account on GitHub.

# CONCLUSION

---

- To Evaluate our model and to achieve a better performance which is done by using XGB classifier.
- This is one of the best optimization technique which is decision-tree based and adopts gradient boosting framework technique for analysis and confusion matrix.
- XG Boost has tremendous predictive power and is about 10 times more durable than other gradient boosting techniques.
- It is further recognized as the “Regularized Boosting” Technique ,like it has true positive,true negative,false positive,false negative values to evaluate the performance of the classification model.