Intelligent Emergency-Auto-Adjustments for Faculty

SECTION: K18ZV

Final Project report:

Registration No.	Name	Roll no:
11812121	Gatla Akanksh	15
11813649	Bhoj Kumar	13
11812075	Abhiraj Singh	16
11801980	Rishabh	14

Department of Intelligent Systems

School of Computer Science Engineering

Lovely Professional University, Jalandhar

April-2020

TABLE OF CONTENTS

TITLE PAGE NO.1

1.Introduction:

As the name infers, Intelligent Emergency auto modifications for Faculty Helps the University to discover the staff on recreation and alter them with the workforce on crisis leave that causes understudy to go with their classes not surprisingly.

How would you recognize Faculty on leave and change them with workforce on relaxation?

1. A University has numerous instructors for the equivalent subject/course. Identify Entities. The substances right now are Department, Teachers accessible.

2. alteration. We developed timetable grids for each hold educators and normal teachers

3. we likewise have the executive's mode with the goal that one can without much of a stretch include more resources by which TensorFlow execution would be conceivable

4. Instructor Priorities, save workforce who are accessible for task can be organized and can be picked the best one for the activity.

2. Background

We use simple code not advance to enforce the mission

"**Intelligent Emergency auto adjustment for faculty**".

In this project just like query management it is ask the information and implement your adjustment with it. The history section should talk your findings in a chronological way to accentuate the development in the discipline and the missing points that need to be addressed. The historical past must be written as a precis of your interpretation of previous studies and what

your look at proposes to accomplish.

3. Motivation

Unlike maximum tangible project management

functions, motivation is not certain by the project supervisor to a group member, alternatively motivation is inner to every team member and derived from a crew member's preference to attain a goal, accomplish a task, or work in the direction of expectations.

4. Goals and objective

our primary and important Goal here turned into to reap this problem of assigning schools to the magnificence that was abandoned by way of their

Professor to reap maximum utilisation of the time of students ans nicely of the colleges and reserve colleges.

We made it our number one objective to make this program easy to use and be scalable and be effortlessly diagnosable in case of any malfunctioning.

Description

this code is of faculty management device and uses AI concepts to assign reserve professors in threat one lively professor applies leave.

5. Technologies and frameworks used

Python:

we selected python as our choice due to its ease of

implementation and simplicity of use and the really nicely integration of the A.I. Libraries in python.

6. Code:

```
import numpy as np

import datetime

import calendar

import pandas as pd


def login_screen(teacherDB):

    print("Auto Adjustment For Faculty")
```

```
    print("Teacher 1: 1021 \n Teacher 2: 1022 \n Teacher 3: 1023 \n Teacher
4: 1024 \n")
    teacherID = int(input("please enter your Teacher ID >> "))
    if(teacherID in teacherDB.keys()):
        teacherID = teacherDB[teacherID]
        welcome_screen(teacherID)
    else:
        print("your record doesnot exist")


def show_free_slots(teacherID, day):
    for time_Slot in range(5):
        if (profTT[teacherID][day][time_Slot] == 0):
            print(trans_dict_time[time_Slot] , " is free")


def show_taken_slots(teacherID, day):
    for time_Slot in range(5):
        if (profTT[teacherID][day][time_Slot] == 1):
            print(trans_dict_time[time_Slot] , " is occupied")


def show_makeup_slots(teacherID, day):
    for time_Slot in range(5):
        if (profTT[teacherID][day][time_Slot] == 2):
```

```
            print(trans_dict_time[time_Slot] , " is make up adjustment")


def welcome_screen(teacherID):
    print("\n\nFaculty Requirment Portal! \n")
    print("please select one option from the list.")
    print("Select 1>    to veiw schedule ")
    print("Select 2>    to appoint a make-up class ")
    print("Select 3>    to apply emergency leave ")
    print("Select 4>    to exit this portal ")

    choice = int(input("option > "))
    if(choice == 1):
        day = input("enter the day >> ")
        print("\nYOUR SCHEDULE \n")
        day = trans_dict_day[day]
        show_taken_slots(teacherID, day)
        show_free_slots(teacherID, day)
        show_makeup_slots(teacherID, day)
        choice = input("would you like to return to home page? >> ").lower()
        if(choice == "yes"):
            print("redirecting you to home page \n\n")
```

```
        welcome_screen(teacherID)


elif(choice == 2):
    day = input("please enter the day you wish to apply for a make up >> ")
    day = trans_dict_day[day]
    print("please choose time slot for the leave")
    print("10:00AM - 11:00AM >> 0")
    print("11:00AM - 12:0PAM >> 1")
    print("1:00PM - 2:00PM >> 2")
    print("2:00PM - 3:00PM >> 3")
    print("3:00PM - 4:00PM >> 4")
    time_slot = int(input("option >> "))
    MakeUpApp(0,teacherID,day,time_slot)


elif(choice == 3):
    day = input("please enter the day you wish to apply for a leave >> ")
    day = trans_dict_day[day]
    print("please choose time slot for the leave")
    print("10:00AM - 11:00AM >> 0")
    print("11:00AM - 12:0PAM >> 1")
    print("1:00PM - 2:00PM >> 2")
    print("2:00PM - 3:00PM >> 3")
```

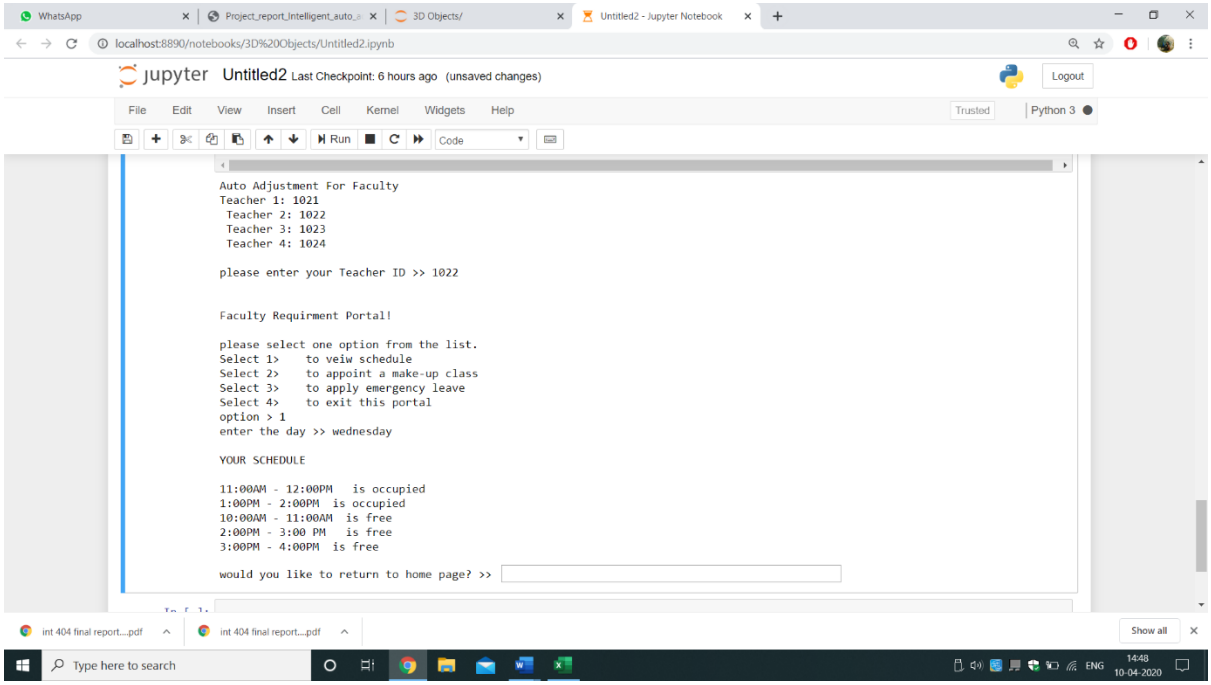
```
        print("3:00PM - 4:00PM >> 4")
        time_slot = int(input("option >> "))
        leaveApp(teacherID,day,time_slot)


    elif(choice == 4):
        print("\nexiting!")


def leaveApp(teacherID, day, time_slot):
    print("\nupdating time schedule")

    profTT[teacherID][day][time_slot] = 0
    MakeUpApp(1,teacherID,day,time_slot)
    choice = input("would you like to return to home page? >> ").lower()
    if(choice == "yes"):
        print("redirecting you to home page \n\n")
        welcome_screen(teacherID)


def MakeUpApp(flag,teacherID,day,time_slot):
    if(flag == 0):
```

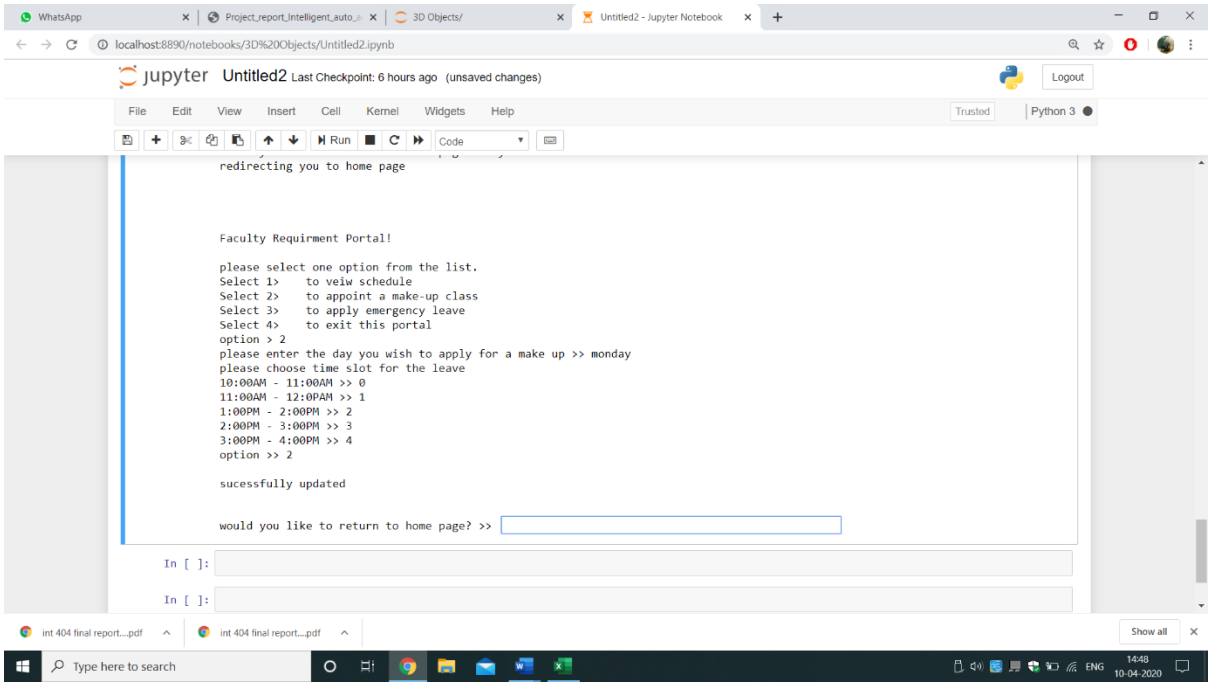
```
        profTT[teacherID][day][time_slot] = 2
        print("\nsucessfully updated\n")
        choice = input("would you like to return to home page? >> ").lower()
        if(choice == "yes"):
            print("redirecting you to home page \n\n")
            welcome_screen(teacherID)
    else:
        ResProfChoices = []
        for ResProf in range(len(resprofTT)):
            if(resprofTT[ResProf][day][time_slot] == 0):
                ResProfChoices.append(ResProf)
        if(ResProfChoices.count == 1):
            resprofTT[ResProfChoices[0]][day][time_slot] = 2
            print("\n\nassigning reserve professor",ResProfChoices[0]+1,"for the
time slot\n\n")
        else:
            #print(ResProfChoices)
            bestChoice = GetBestChoice(ResProfChoices,day,time_slot)
            print("\n\nassigning reserve professor",bestChoice+1,"for the time
slot\n\n")


def GetBestChoice(choices,day,time_slot):
```

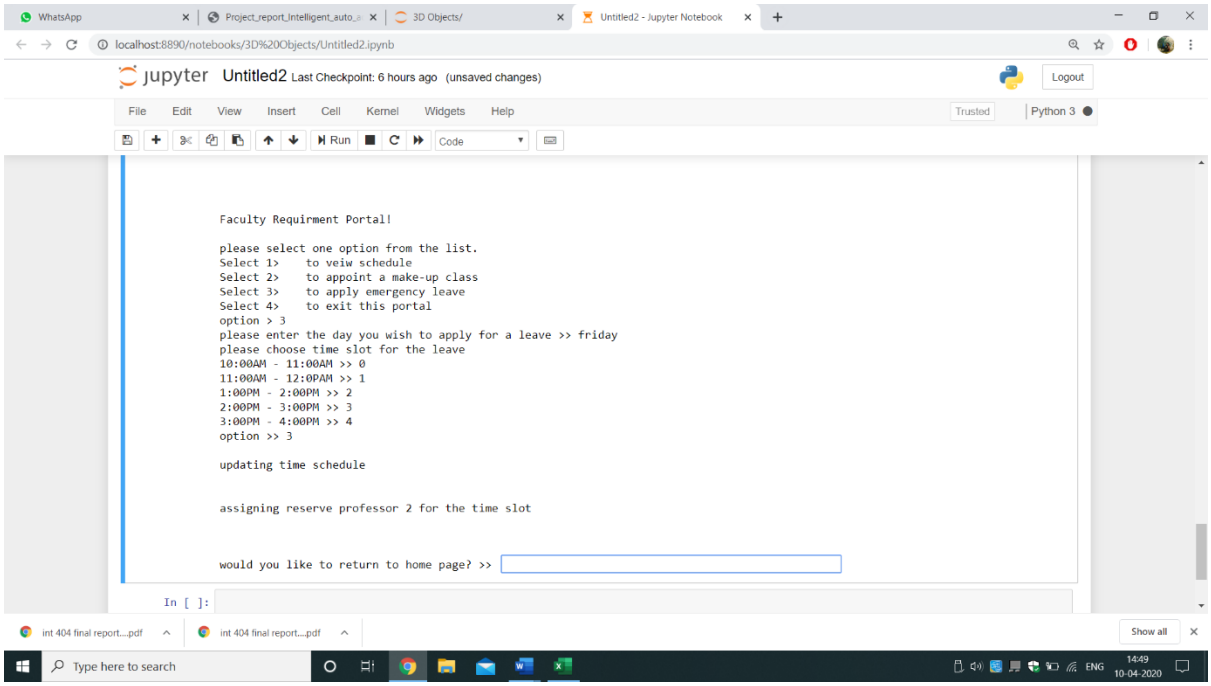
```
    priorityTable = {}
    for i in choices:
        priorityTable[i] = 0
        if(time_slot > 4 and time_slot < 0):
            if (resprofTT[i][day][time_slot-1] == 1 or
resprofTT[i][day][time_slot+1] == 1):
                priorityTable[i] += 1
            elif (resprofTT[i][day][time_slot-1] == 1 and
resprofTT[i][day][time_slot+1] == 1):
                priorityTable[i] += 2
    #print(priorityTable)
    for key,value in priorityTable.items():
            if(value == max(priorityTable.values())):
                #print(key)
                return key
defTT =  [[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]]

profTT = [[[0,1,0,1,0],[1,1,0,0,1],[1,1,0,1,1],[0,1,1,1,0],[1,1,0,1,1]],
          [[1,0,0,1,1],[0,1,0,1,0],[0,1,1,0,0],[1,0,1,0,0],[1,0,1,0,1]],
          [[0,1,1,0,1],[1,0,1,0,0],[1,0,0,1,0],[1,0,1,0,1],[1,1,1,0,0]],
          [[1,1,1,0,0],[0,0,1,1,1],[0,0,1,0,1],[0,1,0,1,1],[0,0,1,1,1]]]
```

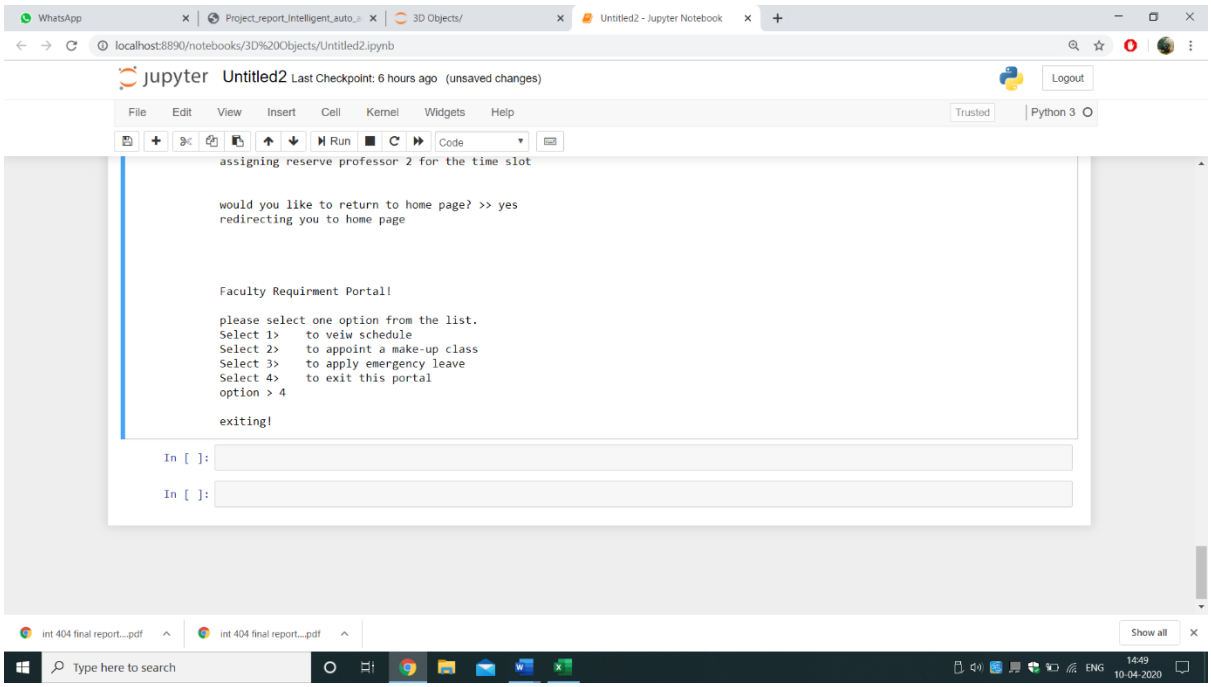
```
resprofTT = [[[1,0,0,1,1],[0,1,0,1,1],[1,1,0,0,1],[1,0,0,1,1],[0,1,0,1,1]],
            [[0,1,1,1,0],[1,1,0,0,0],[0,1,0,1,0],[0,1,1,0,0],[1,1,0,0,0]],
            [[1,1,0,0,0],[1,0,1,0,0],[1,0,0,1,1],[1,1,0,1,1],[1,0,1,0,0]],
            [[0,0,1,1,1],[0,0,1,1,1],[1,0,1,0,0],[0,0,1,0,1],[0,0,1,1,1]]]


trans_dict_day = {"monday" :0, "tuesday":1, "wednesday":2, "thursday":3,
"friday":4}

trans_dict_day_opp = {0:"monday" ,1:"tuesday", 2:"wednesday",
3:"thursday",4: "friday"}

trans_dict_time= {0:"10:00AM - 11:00AM", 1:"11:00AM - 12:00PM ",
2:"1:00PM - 2:00PM", 3:"2:00PM - 3:00 PM ", 4:"3:00PM - 4:00PM"}

teacherDB = {1:0, 1021 :0, 1022 :1, 1023:2, 1024:3}


login_screen(teacherDB)
```

7. Output:


jupyter Untitled2 Last Checkpoint: 6 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Auto Adjustment For Faculty
Teacher 1: 1021
Teacher 2: 1022
Teacher 3: 1023
Teacher 4: 1024
please enter your Teacher ID >> 1022
Faculty Requirment Portal!
please select one option from the list.
Select 1> to veiw schedule
Select 2> to appoint a make-up class
Select 3> to apply emergency leave
Select 4> to exit this portal
option > 1
enter the day >> wednesday
YOUR SCHEDULE
11:00AM - 12:00PM is occupied
1:00PM - 2:00PM is occupied
10:00AM - 11:00AM is free
2:00PM - 3:00 PM is free
3:00PM - 4:00PM is free
would you like to return to home page? >>
Show all
Type here to search
ENG
14:48
10-04-2020


jupyter Untitled2 Last Checkpoint: 6 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
redirecting you to home page
Faculty Requirment Portal!
please select one option from the list.
Select 1> to veiw schedule
Select 2> to appoint a make-up class
Select 3> to apply emergency leave
Select 4> to exit this portal
option > 2
please enter the day you wish to apply for a make up >> monday
please choose time slot for the leave
10:00AM - 11:00AM >> 0
11:00AM - 12:0PAM >> 1
1:00PM - 2:00PM >> 2
2:00PM - 3:00PM >> 3
3:00PM - 4:00PM >> 4
option >> 2
sucessfully updated
would you like to return to home page? >>
In [ ]:
In [ ]:
Show all
Type here to search
ENG
14:48
10-04-2020


WhatsApp
Project_report_Intelligent_auto...
3D Objects/
Untitled2 - Jupyter Notebook
localhost:8890/notebooks/3D%20Objects/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: 6 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Faculty Requirment Portal!
please select one option from the list.
Select 1> to veiw schedule
Select 2> to appoint a make-up class
Select 3> to apply emergency leave
Select 4> to exit this portal
option > 3
please enter the day you wish to apply for a leave >> friday
please choose time slot for the leave
10:00AM - 11:00AM >> 0
11:00AM - 12:0PAM >> 1
1:00PM - 2:00PM >> 2
2:00PM - 3:00PM >> 3
3:00PM - 4:00PM >> 4
option >> 3
updating time schedule
assigning reserve professor 2 for the time slot
would you like to return to home page? >>
In [ ]:
int 404 final report....pdf
int 404 final report....pdf
Show all
Type here to search
ENG
14:49
10-04-2020


WhatsApp
Project_report_Intelligent_auto...
3D Objects/
Untitled2 - Jupyter Notebook
localhost:8890/notebooks/3D%20Objects/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: 6 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
assigning reserve professor 2 for the time slot
would you like to return to home page? >> yes
redirecting you to home page
Faculty Requirment Portal!
please select one option from the list.
Select 1> to veiw schedule
Select 2> to appoint a make-up class
Select 3> to apply emergency leave
Select 4> to exit this portal
option > 4
exiting!
In [ ]:
In [ ]:
int 404 final report....pdf
int 404 final report....pdf
Show all
Type here to search
ENG
14:49
10-04-2020

8. Swot Analysis achieved inside the project

Strengths:

● Unique interface which is simple to apprehend and use

● Can be scaled up nationally and globally.

● may be connected effortlessly to database and then shop the data for authentication

● Relatively clean to diagnose the issues if occured

Weaknesses:

● Not anybody can use this service as it is available to offline database handiest for now

● tideous paintings of adding in colleges

● very sluggish if a variety of faculties time schedules are present

Opportunities:

● Expand excessive the usage of TensorFLow

● can be used by colleges faculties and all educational institutions so we can find customers without problems

Threats:

● feasible authentication failures.

● no fail safe methods added

- no backups can be made to the database

Thank you !