TITLE OF THE PROJECT

Submitted by

Name of the Students: Aritra Ghosal
Enrolment number: 12022002018036
Section: F
Class Roll Number: 28
Stream: C.S.B.S
Subject: Programming for Problem Solving
Subject Code: IVC101
Department: Basic Science and Humanities

Under the supervision of
Name of the teachers

Academic Year: 2022-26

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE FIRST SEMESTER

DEPARTMENT OF BASIC SCIENCE AND HUMANITIES
INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA

CERTIFICATE OF RECOMMENDATION

We hereby recommed that the project prepared under our supervision by Aritra Ghosal, entitled Title of the Project be accepted in partial fulfillment of the requirements for the degree of partial fulfillment of the first semester.

Head of the Department Basic Sciences and Humanities IEM, Kolkata	Project Supervisor

1 Introduction

Python is a versatile and easy to use language often used in data manipulation. What separates Python from all other languages is its large number of use cases. Whereas Javascript is used for the web, C for systems, R for data, Python can be used for all three and many more. The following project demonstrates a model system run using mainly python.

1.1 Objective

This project attempts to model a small scale database management system utilized by an academic institution. The objective of this project is to learn and demonstrate several python programming concepts including:

- Using python code from other files
- Importing and using third party modules
- Reading and writing text files
- Managing CSV data
- Plotting data
- Building a basic user interface
- Utilizing concepts of Object Oriented Programming

This project also demonstrates general programming concepts such as ER diagrams.

1.2 Organization of the Project

```
.
├── code
│   ├── batch.py
│   ├── course.py
│   ├── department.py
│   ├── examination.py
│   ├── main.py
│   └── student.py
├── databases
│   ├── batch.csv
│   ├── course.csv
│   ├── department.csv
│   └── student.csv
├── fonts
│   ├── CONSOLAB.TTF
│   ├── timesnewroman-bold.ttf
│   ├── timesnewroman-bolditalic.ttf
│   ├── timesnewroman-italic.ttf
│   └── timesnewroman-regular.ttf
├── instructions
│   ├── IEM logo.png
│   ├── Project Details.docx
│   └── Project Report Template.docx
├── latex
│   ├── er-diagram.sty
│   ├── er.tex
│   ├── outputs.tex
│   ├── project.tex
│   └── template.tex
├── Makefile
└── outputs
    ├── output.log
    └── …
```

The **code** directory contains all the python code that is being executed at runtime. **batch.py** is a module that exports functions that operate on a batch. Likewise, **course.py** is a module that exports functions that operate on courses in the database. Same for **department.py**, which is a module that exports functions that operate on a department. **examination.py** exports the **Examination** class that represents an examination being held by the institution. **main.py** is a file with executive permissions which imports all of the above and runs a simple menu based command line user interface.

The **databases** directory contains all the data in CSV format.

The **fonts** directory contains the fonts required to compile this document.

The **instructions** directory contains all of the raw material to given to build this project.

The **latex** directory contains all of the LaTeX code used to build the project report (this file). **template.tex** sets the default values necessary for the project report. **project.tex** contains the code that is compiled into the project report. It contains sources the outputs and diagrams along with the python code to include in the project report.
er.tex contains the er diagram for the database and **er-diagram.sty** is a third party library used to draw the er diagram. **output.tex** is an automatically generated file which sources all of the plots into the final report.

The **Makefile** contains the build system for the entire project. It specifies the dependencies for each component and runs the commands to create each component. The **Makefile** also contains code that generates the databases and fills them with random data modelling the system as closely as possible. This is the centre point of the entire project, it determines the order and execution of everything else in the project.

The **outputs** directory contains all of the output generated by the python code at runtime. The **output.log** file is generated file running the python code, it contains the entire interaction between the program and the user via the command line interface and stores it for future reference.

2 Database Descriptions

Each student in the **student.csv** database has a unique ID, along with a name and a class roll number. Each student is associated with a single batch.
Each batch in **batch.csv** is assigned a unique ID. They also have name and a department they fall under. Each batch has a list of courses and a list of students who appear for the courses.
Each course in **course.csv** has an ID, subject name and a storage of marks obtained by each student appearing for the course.
Each department in **department.csv** has an ID, name and list of batches that worked under that department.

2.1 Database Samples

batch.csv

Batch ID	Batch Name	Department Name	List of Courses	List of Students
CSE00	CSE 2000-2004	CSE	…	…
CSE05	CSE 2005-2009	CSE	…	…
CSE06	CSE 2006-2010	CSE	…	…
CSE08	CSE 2008-2012	CSE	…	…
CSE09	CSE 2009-2013	CSE	…	…

CSE14	CSE 2014-2018	CSE	…	…
CSE90	CSE 1990-1994	CSE	…	…
CSE92	CSE 1992-1996	CSE	…	…
CSE93	CSE 1993-1997	CSE	…	…
CSE95	CSE 1995-1999	CSE	…	…
CSE96	CSE 1996-2000	CSE	…	…
CSE98	CSE 1998-2002	CSE	…	…
ECE00	ECE 2000-2004	ECE	…	…
ECE01	ECE 2001-2005	ECE	…	…
ECE02	ECE 2002-2006	ECE	…	…
ECE03	ECE 2003-2007	ECE	…	…
ECE08	ECE 2008-2012	ECE	…	…
ECE13	ECE 2013-2017	ECE	…	…
ECE15	ECE 2015-2019	ECE	…	…
ECE18	ECE 2018-2022	ECE	…	…
ECE21	ECE 2021-2025	ECE	…	…
ECE89	ECE 1989-1993	ECE	…	…
ECE91	ECE 1991-1995	ECE	…	…
ECE92	ECE 1992-1996	ECE	…	…
ECE94	ECE 1994-1998	ECE	…	…
ECE95	ECE 1995-1999	ECE	…	…
ECE97	ECE 1997-2001	ECE	…	…
ECE98	ECE 1998-2002	ECE	…	…
ECE99	ECE 1999-2003	ECE	…	…
IT03	IT 2003-2007	IT	…	…
IT05	IT 2005-2009	IT	…	…
IT08	IT 2008-2012	IT	…	…
IT09	IT 2009-2013	IT	…	…
IT11	IT 2011-2015	IT	…	…
IT14	IT 2014-2018	IT	…	…
IT91	IT 1991-1995	IT	…	…
IT92	IT 1992-1996	IT	…	…
IT96	IT 1996-2000	IT	…	…
IT99	IT 1999-2003	IT	…	…
CSE22	CSE 2022-2026	CSE	…	…

course.csv

Course ID	Course Name	Marks Obtained
CSE00	CSE 2000-2004	CSE
CSE05	CSE 2005-2009	CSE
CSE06	CSE 2006-2010	CSE
CSE08	CSE 2008-2012	CSE
CSE09	CSE 2009-2013	CSE
CSE14	CSE 2014-2018	CSE
CSE90	CSE 1990-1994	CSE
CSE92	CSE 1992-1996	CSE
CSE93	CSE 1993-1997	CSE
CSE95	CSE 1995-1999	CSE
CSE96	CSE 1996-2000	CSE
CSE98	CSE 1998-2002	CSE
ECE00	ECE 2000-2004	ECE
ECE01	ECE 2001-2005	ECE
ECE02	ECE 2002-2006	ECE
ECE03	ECE 2003-2007	ECE
ECE08	ECE 2008-2012	ECE
ECE13	ECE 2013-2017	ECE
ECE15	ECE 2015-2019	ECE
ECE18	ECE 2018-2022	ECE
ECE21	ECE 2021-2025	ECE
ECE89	ECE 1989-1993	ECE
ECE91	ECE 1991-1995	ECE
ECE92	ECE 1992-1996	ECE
ECE94	ECE 1994-1998	ECE
ECE95	ECE 1995-1999	ECE
ECE97	ECE 1997-2001	ECE
ECE98	ECE 1998-2002	ECE
ECE99	ECE 1999-2003	ECE
IT03	IT 2003-2007	IT
IT05	IT 2005-2009	IT
IT08	IT 2008-2012	IT
IT09	IT 2009-2013	IT
IT11	IT 2011-2015	IT
IT14	IT 2014-2018	IT
IT91	IT 1991-1995	IT

IT92	IT 1992-1996	IT
IT96	IT 1996-2000	IT
IT99	IT 1999-2003	IT
CSE22	CSE 2022-2026	CSE

`department.csv`

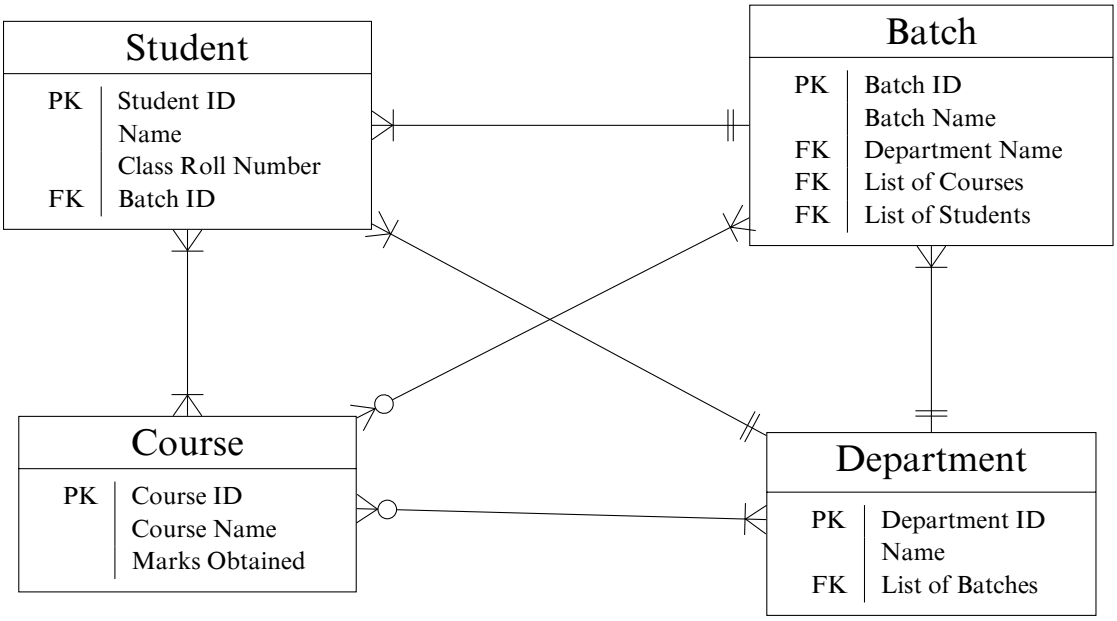
Department ID	Department Name	List of Batches
CSE	Computer Science and Engineering	…
ECE	Electronics and Communication Engineering	…
IT	Information Technology	…
BA	Business Administration	…

`student.csv`

Student ID	Name	Class Roll No	Batch ID
ECE1531	Adah Soman	C-45	ECE15
CSE0679	Zain Dora	F-89	CSE06
ECE8925	Mahika Gole	H-53	ECE89
IT1431	Manjari Trivedi	D-65	IT14
ECE1518	Kabir Andra	D-18	ECE15
CSE0551	Mohanlal Wali	D-92	CSE05
ECE9959	Samiha Balakrishnan	C-45	ECE99
CSE9588	Devansh Bora	H-09	CSE95
ECE0262	Kabir Chahal	E-31	ECE02
ECE2148	Vardaniya Kar	G-66	ECE21
ECE9805	Dishani Dass	G-73	ECE98
ECE9418	Tushar Hans	D-57	ECE94
ECE9264	Ayesha Sangha	G-48	ECE92
CSE9640	Nirvi Thaker	B-07	CSE96
IT0589	Mishti Mall	C-31	IT05
IT1194	Divyansh Aggarwal	B-64	IT11
IT1473	Kabir Sandhu	C-09	IT14
CSE0819	Siya Chopra	G-73	CSE08
IT9142	Advik Sant	C-59	IT91
CSE0005	Anya Bala	G-59	CSE00
ECE1330	Sara Gade	E-95	ECE13

ECE2195	Onkar Lata	G-53	ECE21
CSE9381	Nakul Varma	E-61	CSE93
IT0836	Madhav Gour	H-56	IT08
CSE0956	Rasha Varma	B-53	CSE09
ECE9531	Tara Krishnan	D-42	ECE95
IT0931	Biju Wali	B-81	IT09
CSE9670	Uthkarsh Lal	G-50	CSE96
ECE0239	Arhaan Tiwari	A-80	ECE02
ECE1384	Zain Taneja	F-62	ECE13
CSE9098	Piya Karnik	G-06	CSE90
ECE0309	Zaina Bava	D-53	ECE03
IT9279	Hazel Mani	F-71	IT92
IT0370	Dishani Din	F-48	IT03
CSE0018	Nirvi Varkey	A-42	CSE00
ECE9945	Tanya Keer	E-63	ECE99
ECE9113	Emir Dass	H-18	ECE91
CSE9855	Kavya Warrior	G-45	CSE98
ECE0142	Oorja Bajwa	E-13	ECE01
ECE9739	Ranbir Thakkar	B-68	ECE97
IT9966	Ritvik Dey	B-02	IT99
ECE0818	Diya Mann	C-07	ECE08
IT9681	Adah Madan	F-40	IT96
ECE1834	Urvi Datta	C-77	ECE18
ECE9466	Vedika Kapur	H-86	ECE94
ECE0026	Hridaan Kibe	D-36	ECE00
CSE1417	Arnav Gade	G-87	CSE14
IT9968	Alisha Kala	B-59	IT99
ECE1873	Nishith Kala	F-18	ECE18
CSE9242	Yuvaan Ganesan	A-63	CSE92
IT0998	Vritika Varghese	C-45	IT09
ECE9758	Priyansh Rege	C-91	ECE97
CSE0948	Kartik Joshi	B-22	CSE09

3 E-R Diagram



4 Programs

main.py

```
#!/bin/python3
from re import search
#import from modules
from student import
 ↪  create_student,update_student,remove_student,report
from course import create_course,course_performance,course_statistics
from batch import
 ↪  create_batch,students,courses,batch_performance,batch_statistics
from department import
 ↪  create_department,batches,batch_averages,department_statistics
from examination import Examination
def input_marks():
    while True:
        roll_number=input('\n\t\t\tClass Roll Number: ')
        if roll_number=='':
            break
        yield {
            'roll number':roll_number,
            'name':input('\t\t\tStudent Name: '),
            'marks':float(input('\t\t\tMarks: '))
        }
def input_array(data,id):
    print(f'\t\t\tEnter the {data} for {id}')
    while True:
        data=input('\t\t\t\t: ')
        if data=='':break
        yield data
while True:
    choice=input('''
```

```
1. Student
2. Course
3. Batch
4. Department
5. Examination
: ''')
    if choice=='':break
    elif choice=='1':
        choice=input('''
    1. Create a new student
    2. Update details of a student
    3. Remove a student
    4. Generate report of a student
    : ''')
        if choice=='1':
            create_student(
                student_id=input('\t\tStudent ID: '),
                name=input('\t\tStudent Name: '),
                class_roll_no=input('\t\tClass Roll No: '),
                batch=input('\t\tBatch ID: ')
            )
        elif choice=='2':
            update_student(
                student_id=input('\t\tStudent ID: '),
                name=input('\t\tStudent Name: '),
                class_roll_no=input('\t\tClass Roll No: '),
            )
        elif choice=='3':
            remove_student(
                student_id=input('\t\tStudent ID: ')
            )
        elif choice=='4':
            report(
                student_id=input('\t\tStudent ID: ')
            )
    elif choice=='2':
        choice=input('''
    1. Create a new course
    2. View performance of all students
    3. Create course statistics
    : ''')
        if choice=='1':
            create_course(
                course_id=input('\t\tCourse ID: '),
                course_name=input('\t\tCourse Name: '),
                marks=[student for student in input_marks()]
            )
        elif choice=='2':
            course=input('\t\tCourse: ')
            if search('^C0[0-9]{2}$',course):
                for i in course_performance(course_id=course):
                    print('\t\t\t',i)
            else:
                for i in course_performance(course_name=course):
                    print('\t\t\t',i)
        elif choice=='3':
            course=input('\t\tCourse: ')
            if search('^C0[0-9]{2}$',course):
                course_statistics(course_id=course)
            else:
```

```
                course_statistics(course_name=course)
    elif choice=='3':
        choice=input('''
    1. Create a new batch
    2. View list of students in a batch
    3. View list of courses taught in a batch
    4. View performance of a batch
    5. Create pie chart of percentage of all students
    : ''')
        batch_id=input('\t\tBatch ID: ')
        if choice=='1':
            create_batch(
                batch_id=batch_id,
                batch_name=input('\t\tBatch Name: '),
                department_name=input('\t\tDepartment Name: '),
                courses=[i for i in input_array('courses',batch_id)],
                students=[i for i in input_array('students',batch_id)]
            )
        elif choice=='2':
            print('\t\t',students(batch_id=batch_id))
        elif choice=='3':
            print('\t\t\t',courses(batch_id=batch_id))
        elif choice=='4':
            for i in
             ↪  batch_performance(batch_id=batch_id):print('\t\t\t',i)
        elif choice=='5':
            batch_statistics(batch_id=batch_id)
    elif choice=='4':
        choice=input('''
    1. Create a new department
    2. View batches of a department
    3. View average performance of batches of a department
    4. Create statistics of a department
    :''')
        department_id=input('\t\tDepartment ID: ')
        if choice=='1':
            create_department(
                department_id=department_id,
                department_name=input('\t\tDepartment Name: '),
                batches=[i for i in
                 ↪  input_array('batches',department_id)]
            )
        elif choice=='2':
            print('\t\t\t',batches(department_id=department_id))
        elif choice=='3':
            for i in batch_averages(department_id=department_id):
                print(i)
        elif choice=='4':
            department_statistics(department_id=department_id)
    elif choice=='5':
        print('''
    Hold an examination:
    ''')
        exam=Examination(*[i for i in input_array('batches','exam')])
        choice=input('''
    1. View student perfomance in the examination
    2. Create examination statistics
    : ''')
        if choice=='1':
```

```
            print(exam.student_performance)
        elif choice=='2':
            exam.statistics()
```

student.py

```
from csv import writer,reader
from texttable import Texttable
def create_student(**kwargs):
    batch_id=kwargs['batch']
    student_id=kwargs['student_id']
    with open('databases/student.csv','a') as csvfile:
        writer(csvfile).writerow([
            student_id,
            kwargs['name'],
            kwargs['class_roll_no'],
            batch_id
        ])
    rows=[]
    with open('databases/batch.csv','r') as csvfile:
        for row in reader(csvfile):
            if row[0]==batch_id:
                row[4]+=f':{student_id}'
            rows.append(row)
    with open('databases/batch.csv','w') as csvfile:
        db=writer(csvfile)
        for row in rows:
            db.writerow(row)
def update_student(**kwargs):#update by student id
    rows=[]
    EXIT_CODE=1
    with open('databases/student.csv','r') as csvfile:
        db=reader(csvfile)
        for row in db:
            if row[0]==kwargs['student_id']:
                EXIT_CODE=0
                rows.append([
                    row[0],
                    kwargs['name'] if 'name' in kwargs else row[1],
                    kwargs['class_roll_no'] if 'class_roll_no' in
                     ↪  kwargs else row[2],
                    kwargs['student_id'][:-2]
                ])
                break
            rows.append(row)
        for row in db:rows.append(row)#add remaining
    with open('databases/student.csv','w') as csvfile:#update file
        db=writer(csvfile)
        for row in rows:db.writerow(row)
    return EXIT_CODE
def remove_student(student_id):#remove by student id
    rows=[]
    EXIT_CODE=1
    with open('databases/student.csv','r') as csvfile:
        db=reader(csvfile)
```

```
    for row in db:
        if row[0]==student_id:#found
            batch_id=row[3]
            EXIT_CODE=0
            break
        rows.append(row)
    for row in db:rows.append(row)#add remaining
with open('databases/student.csv','w') as csvfile:#update file
    db=writer(csvfile)
    for row in rows:db.writerow(row)
if EXIT_CODE==1:return 1#student not found
rows=[]
empty_batch=False
with open('databases/batch.csv','r') as csvfile:
    db=reader(csvfile)
    for row in db:
        if row[0]==batch_id:
            students=row[4].split(':')
            students.remove(student_id)
            courses=row[3].split(':')
            if len(students)==0:
                empty_batch=True
                department_name=row[2]
            else:
                row[4]=':'.join(students)
                rows.append(row)
            break
        rows.append(row)
    for row in db:rows.append(row)
with open('databases/batch.csv','w') as csvfile:
    db=writer(csvfile)
    for row in rows:db.writerow(row)
rows=[]
with open('databases/course.csv','r') as csvfile:
    db=reader(csvfile)
    for row in db:
        if row[0] in courses:
            marks=row[2]
            a=marks.index(student_id)
            b=marks.find('-',a)
            row[2]=marks[:a-1]+marks[b:]
        rows.append(row)
with open('databases/course.csv','w') as csvfile:
    db=writer(csvfile)
    for row in rows:db.writerow(row)
if not empty_batch:return 0
rows=[]
with open('databases/department.csv','r') as csvfile:
    db=reader(csvfile)
    for row in db:
        if row[0]==department_name:
            batches=row[2].split(':')
            batches.remove(batch_id)
            row[2]=':'.join(batches)
            rows.append(row)
            break
        rows.append(row)
```

```
        for row in db:rows.append(row)
    with open('databases/department.csv','w') as csvfile:
        db=writer(csvfile)
        for row in rows:db.writerow(row)
def report(student_id):
    def grade(marks):
        if marks>=90:grade='A'
        elif marks>=80:grade='B'
        elif marks>=70:grade='C'
        elif marks>=60:grade='D'
        elif marks>=50:grade='E'
        else: return 'F','Failed'
        return (grade,'Passed')
    EXIT_CODE=1
    with open('databases/student.csv') as csvfile:
        db=reader(csvfile)
        for row in db:
            if row[0]==student_id:
                _,name,roll,batch_id=row
                EXIT_CODE=0
                break
    if EXIT_CODE==1:return 1
    with open('databases/batch.csv') as csvfile:
        db=reader(csvfile)
        for row in db:
            if row[0]==batch_id:
                exams=row[3].split(':')
                break
    marksheet=Texttable()
    marksheet.set_cols_align(('l','l','r','r','c','l'))
    marksheet.add_row(['Course','Course Id','Marks Obtained','Full
     ↪  Marks','Grade','Remarks'])
    total=0
    with open('databases/course.csv') as csvfile:
        db=reader(csvfile)
        for row in db:
            if row[0] in exams:
                performance=row[2]
                i=performance.index(student_id)
                a=performance.find(':',i)
                b=performance.find('-',i)
                marks=float(performance[a+1:b])
                total+=marks
                marksheet.add_row([
                    row[1],
                    row[0],
                    marks,
                    100,
                    *grade(marks)
                ])
    number=len(exams)
    marksheet.add_row(['Total','-',total,number*100,*grade(total/numb
     ↪  er)])
    with open(f'outputs/{student_id}-report_card.txt','w') as
     ↪  report:report.write(f'''
{name} ({roll})
{marksheet.draw()}
```

```
ID:{student_id}
Batch:{batch_id}
 ''')
    return EXIT_CODE
```

course.py

```
from csv import reader,writer
from collections import namedtuple
from matplotlib.pyplot import
 ↪  hist,title,xlabel,ylabel,xticks,xlim,style,close,savefig
def _parse_args(argdict):
    wrong_arg=Exception('Either provide course_id or course_name')
    if len(argdict)>1:raise wrong_arg
    (param,val),=argdict.items()
    if param=='course_id':rown=0
    elif param=='course_name':rown=1
    else:raise wrong_arg
    return rown,val
def create_course(**kwargs):
    marks=''
    batches=set()
    course_id=kwargs['course_id']
    with open('databases/student.csv','r') as csvfile:
        db=reader(csvfile)
        for student_data in kwargs['marks']:
            roll=student_data['roll number']
            for row in db:
                if row[2]==roll:
                    student_id=row[0]
                    marks+=f"{student_id}:{student_data['marks']}-"
                    batches.add(student_id[0:-2])
                    csvfile.seek(0)
                    break
    with open('databases/course.csv','a') as csvfile:
        writer(csvfile).writerow([
            course_id,
            kwargs['course_name'],
            marks[:-1]#skip last '-'
        ])
    rows=[]
    with open('databases/batch.csv','r') as csvfile:
        for row in reader(csvfile):
            if row[0] in batches:
                row[3]+=':'+course_id
            rows.append(row)
    with open('databases/batch.csv','w') as csvfile:
        db=writer(csvfile)
        for row in rows:db.writerow(row)
def course_performance(**kwargs):
    rown,val=_parse_args(kwargs)
    Student=namedtuple("Student",('roll','name','marks'))
    marks=False
    with open('databases/course.csv','r') as csvfile:
        for row in reader(csvfile):
```

```
            if row[rown]==val:
                marks=row[2].split('-')
                break
    if not marks:return -1
    with open('databases/student.csv','r') as csvfile:
        db=reader(csvfile)
        for perf in marks:
            a=perf.index(':')
            student_id=perf[:a]
            for row in db:
                if row[0]==student_id:
                    yield Student(row[2],row[1],float(perf[a+1:]))
                    csvfile.seek(0)#start from beginning
                    break
def course_statistics(**kwargs):
    rown,val=_parse_args(kwargs)
    marks=False
    with open('databases/course.csv','r') as csvfile:
        for row in reader(csvfile):
            if row[rown]==val:
                performance=row[2]
                if performance=='':return -1
                marks=[float(i[i.index(':')+1:]) for i in
                 ↪  performance.split('-')]
                break
    if not marks:return -1
    style.use('Solarize_Light2')
    hist(marks,bins=[0,50,60,70,80,90,100])
    title(val)
    xlabel('marks')
    ylabel('number of students')
    xticks([25,55,65,75,85,95],['F','E','D','C','B','A'])
    xlim(100,0)
    savefig(f'outputs/Course Statistics-{val}.pdf')
    close()
```

batch.py

```
from csv import reader,writer
from functools import partial
from collections import namedtuple
from matplotlib.pyplot import
 ↪  pie,title,style,xticks,yticks,close,savefig
Student=namedtuple("Student",('roll','name','percentage'))
def _parse_args(argdict):
    wrong_arg=Exception('Either provide batch_id or batch_name')
    if len(argdict)>1:raise wrong_arg
    (param,val),=argdict.items()
    if param=='batch_id':rown=0
    elif param=='batch_name':rown=1
    else:raise wrong_arg
    return rown,val
def _direct_list(col,**kwargs):
    rown,val=_parse_args(kwargs)
    with open('databases/batch.csv','r') as csvfile:
        for row in reader(csvfile):
```

```
            if row[rown]==val:
                return row[col].split(':')
    return -1
def create_batch(**kwargs):
    with open('databases/batch.csv','a') as csvfile:
        writer(csvfile).writerow([
            kwargs['batch_id'],
            kwargs['batch_name'],
            kwargs['department_name'],
            ':'.join(kwargs['courses']),
            ':'.join(kwargs['students'])
        ])
students=partial(_direct_list,4)
courses=partial(_direct_list,3)
def batch_performance(**kwargs):
    rown,val=_parse_args(kwargs)
    students=[];exams=[]
    with open('databases/batch.csv','r') as csvfile:
        for row in reader(csvfile):
            if row[rown]==val:
                students=row[4].split(':')
                exams=row[3].split(':')
                break
    if not students and not exams:return -1
    lexams=len(exams)
    with open('databases/student.csv','r') as
     ↪  studentcsv,open('databases/course.csv') as csvfile:
        courses=reader(csvfile)
        for row in reader(studentcsv):
            student_id=row[0]
            if student_id in students:
                total=0
                for course in courses:
                    if course[0] in exams:
                        marks=course[2]
                        i=marks.index(student_id)
                        a=marks.find(':',i)
                        b=marks.find('-',i)
                        total+=float(marks[a+1:b])
                csvfile.seek(0)
                yield Student(row[2],row[1],total/lexams)
def batch_statistics(**kwargs):
    slices,roll_numbers=[],[]
    for student in batch_performance(**kwargs):
        slices.append(student.percentage)
        roll_numbers.append(student.roll)
    name=tuple(kwargs.values())[0]
    title(name)
    xticks([],[])
    yticks([],[])
    style.use('Solarize_Light2')
    pie(slices,labels=roll_numbers,shadow=True,frame=True)
    savefig(f'outputs/Batch Statistics-{name}.pdf')
    close()
```

```
from csv import reader,writer
from collections import namedtuple
from matplotlib.pyplot import
 ↪  plot,xlabel,ylabel,style,title,close,savefig
Batch=namedtuple('Performance',('batch','average'))
def _parse_args(argdict):
    wrong_arg=Exception('Either provide department_id or
     ↪  department_name')
    if len(argdict)>1:raise wrong_arg
    (param,val),=argdict.items()
    if param=='department_id':rown=0
    elif param=='department_name':rown=1
    else:raise wrong_arg
    return rown,val
def create_department(**kwargs):
    with open('databases/department.csv','a') as db:
        writer(db).writerow([
            kwargs['department_id'],
            kwargs['department_name'],
            ':'.join(kwargs['batches'])
        ])
def batches(**kwargs):
    rown,val=_parse_args(kwargs)
    with open('databases/department.csv','r') as db:
        for row in reader(db):
            if row[rown]==val:
                return row[2].split(':')
    return -1
def batch_averages(**kwargs):
    with open('databases/batch.csv','r') as
     ↪  batch_csv,open('databases/course.csv','r') as course_csv:
        batch_db=reader(batch_csv)
        course_db=reader(course_csv)
        for batch in batches(**kwargs):
            total=0
            for row in batch_db:
                if row[0]==batch:
                    batch_csv.seek(0)
                    courses=row[3].split(':')
                    students=row[4].split(':')
                    batch_csv.seek(0)
                    break
            for course in courses:
                for row in course_db:
                    if row[0]==course:
                        performance=row[2]
                        for student in students:
                            i=performance.index(student)
                            a=performance.find(':',i)
                            b=performance.find('-',i)
                            total+=float(performance[a+1:b])
                        course_csv.seek(0)
                        break
            yield Batch(batch,total/(len(students)*len(courses)))
def department_statistics(**kwargs):
```

```
    def year(performance):
        a=float(performance.batch[-2:])
        if a>22:
            return 1900+a
        return 2000+a
    stat=list(batch_averages(**kwargs))
    stat.sort(key=year)
    style.use('Solarize_Light2')
    plot([p.average for p in stat],[p.batch for p in
     ↪  stat],linestyle='--')
    xlabel('Batch Average')
    ylabel('Batch')
    name=tuple(kwargs.values())[0]
    title(name)
    savefig(f'outputs/Department Statistics-{name}.pdf')
    close()
```

examination.py

```
from csv import reader,writer
from numpy import nan,linspace
from collections import namedtuple
from matplotlib.pyplot import
 ↪  scatter,title,xlabel,ylabel,style,legend,close,savefig
from matplotlib.cm import Oranges as colormap #change to change
 ↪  colormap
Student=namedtuple('Performance',('student_id','average'))
class Examination:
    def __init__(self,*batches):
        self.name=input('Name of examination : ')
        exam_data={}
        course_name={}
        #remember data
        with open('databases/course.csv','r') as csvfile:
            csvfile.readline()
            for course_id,name,performance in reader(csvfile):
                exam_data[course_id]={} if performance=='' else
                    dict((i.split(':') for i in
                     ↹  performance.split('-')))
                course_name[course_id]=name
        self.batches=batches
        plot_data={}
        #input data
        self.student_performance=[]
        with open('databases/batch.csv','r') as
         ↪  batchcsv,open('databases/student.csv') as studentcsv:
            student_info=reader(studentcsv)
            for row in reader(batchcsv):
                batch_id=row[0]
                if batch_id in batches:
                    print(batch_id)
                    courses=row[3].split(':')
                    lcourses=len(courses)
                    students=row[4].split(':')
                    lstudents=len(students)
```

```
                    for student in students:
                        total=0
                        for info in student_info:
                            if info[0]==student:#found student id
                                print(f'\t{info[2]}')#print roll
                                 ↪  number
                                studentcsv.seek(0)
                                break
                        for course in courses:
                            entered=input(f'\t\t{course}: ')
                            marks=0 if entered=='' else float(entered)
                            total+=marks
                            exam_data[course][student]=marks
                            try:
                                plot_data[course][batch_id]+=marks/(l
                                 ↪  courses*lstudents)
                            except KeyError:
                                try:
                                    plot_data[course][batch_id]=marks
                                     ↪  /(lcourses*lstudents)
                                except KeyError:
                                    plot_data[course]={batch_id:marks
                                     ↪  /(lcourses*lstudents)}
                        self.student_performance.append(Student(stude
                         ↪  nt,total/lcourses))
        #save data
        with open('databases/course.csv','w') as csvfile:
            db=writer(csvfile)
            db.writerow(['Course ID','Course Name','Marks Obtained'])
            for course in course_name:
                db.writerow([
                    course,
                    course_name[course],
                    '-'.join((f'{student}:{marks}' for student,marks
                     ↪  in exam_data[course].items()))
                ])
        #arrange data
        self.data=[]
        self.courses=[]
        for course,course_data in plot_data.items():
            batch_data=[]
            for batch in batches:
                try:
                    batch_data.append(course_data[batch])
                except KeyError:
                    batch_data.append(nan)
            self.courses.append(course)
            self.data.append(batch_data)
    def statistics(self):
        style.use('Solarize_Light2')
        xlabel('Average Marks')
        ylabel('Batch')
        title(self.name)
        legend(
```

```
            (scatter(marks,self.batches,color=color,edgecolor='black'
                ) for marks,color in
                zip(self.data,colormap(linspace(0,1,len(self.data))))
                ),
            self.courses
        )
        savefig(f'outputs/{self.name} Exam.pdf')
        close()
```

5 Outputs

Command Line Interface

```
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 1
    1. Create a new student
    2. Update details of a student
    3. Remove a student
    4. Generate report of a student
    : 1
                Student ID: CSE0948
                Student Name: Kartik Joshi
                Class Roll No: B-22
                Batch ID: CSE09
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 1
    1. Create a new student
    2. Update details of a student
    3. Remove a student
    4. Generate report of a student
    : 2
                Student ID: ECE9531
                Student Name: Tara Krishnan
                Class Roll No: D-42
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 1
    1. Create a new student
    2. Update details of a student
    3. Remove a student
    4. Generate report of a student
    : 3
                Student ID: IT0072
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 1
    1. Create a new student
    2. Update details of a student
    3. Remove a student
```

```
    4. Generate report of a student
    : 4
                Student ID: ECE1330
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 2
    1. Create a new course
    2. View performance of all students
    3. Create course statistics
    : 1
                Course ID: C011
                Course Name: Robotics
                        Class Roll Number: G-59
                        Student Name: Anya Bala
                        Marks: 89
                        Class Roll Number: A-42
                        Student Name: Nirvi Varkey
                        Marks: 94
                        Class Roll Number:
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 2
    1. Create a new course
    2. View performance of all students
    3. Create course statistics
    : 2
                Course: SDP
                        Student(roll='G-59', name='Anya Bala',
                         ↪  marks=96.0)
                        Student(roll='A-42', name='Nirvi Varkey',
                         ↪  marks=44.0)
                        Student(roll='D-92', name='Mohanlal Wali',
                         ↪  marks=57.0)
                        Student(roll='F-89', name='Zain Dora',
                         ↪  marks=80.0)
                        Student(roll='G-73', name='Siya Chopra',
                         ↪  marks=30.0)
                        Student(roll='B-53', name='Rasha Varma',
                         ↪  marks=72.0)
                        Student(roll='G-87', name='Arnav Gade',
                         ↪  marks=95.0)
                        Student(roll='G-06', name='Piya Karnik',
                         ↪  marks=67.0)
                        Student(roll='A-63', name='Yuvaan Ganesan',
                         ↪  marks=82.0)
                        Student(roll='E-61', name='Nakul Varma',
                         ↪  marks=99.0)
                        Student(roll='H-09', name='Devansh Bora',
                         ↪  marks=65.0)
                        Student(roll='B-07', name='Nirvi Thaker',
                         ↪  marks=34.0)
                        Student(roll='G-50', name='Uthkarsh Lal',
                         ↪  marks=38.0)
                        Student(roll='G-45', name='Kavya Warrior',
                         ↪  marks=46.0)
```

```
Student(roll='D-36', name='Hridaan Kibe',
↪  marks=74.0)
Student(roll='E-13', name='Oorja Bajwa',
↪  marks=73.0)
Student(roll='A-80', name='Arhaan Tiwari',
↪  marks=85.0)
Student(roll='E-31', name='Kabir Chahal',
↪  marks=79.0)
Student(roll='D-53', name='Zaina Bava',
↪  marks=88.0)
Student(roll='C-07', name='Diya Mann',
↪  marks=68.0)
Student(roll='E-95', name='Sara Gade',
↪  marks=90.0)
Student(roll='F-62', name='Zain Taneja',
↪  marks=51.0)
Student(roll='D-18', name='Kabir Andra',
↪  marks=61.0)
Student(roll='C-45', name='Adah Soman',
↪  marks=65.0)
Student(roll='C-77', name='Urvi Datta',
↪  marks=80.0)
Student(roll='F-18', name='Nishith Kala',
↪  marks=66.0)
Student(roll='G-66', name='Vardaniya Kar',
↪  marks=98.0)
Student(roll='G-53', name='Onkar Lata',
↪  marks=53.0)
Student(roll='H-53', name='Mahika Gole',
↪  marks=48.0)
Student(roll='H-18', name='Emir Dass',
↪  marks=65.0)
Student(roll='G-48', name='Ayesha Sangha',
↪  marks=99.0)
Student(roll='D-57', name='Tushar Hans',
↪  marks=64.0)
Student(roll='H-86', name='Vedika Kapur',
↪  marks=85.0)
Student(roll='D-42', name='Tara Krishnan',
↪  marks=82.0)
Student(roll='B-68', name='Ranbir Thakkar',
↪  marks=92.0)
Student(roll='C-91', name='Priyansh Rege',
↪  marks=79.0)
Student(roll='G-73', name='Dishani Dass',
↪  marks=94.0)
Student(roll='E-63', name='Tanya Keer',
↪  marks=60.0)
Student(roll='C-45', name='Samiha
↪  Balakrishnan', marks=38.0)
Student(roll='F-48', name='Dishani Din',
↪  marks=66.0)
```

```
                        Student(roll='C-31', name='Mishti Mall',
                        ↪  marks=46.0)
                        Student(roll='H-56', name='Madhav Gour',
                        ↪  marks=90.0)
                        Student(roll='B-81', name='Biju Wali',
                        ↪  marks=100.0)
                        Student(roll='C-45', name='Vritika
                        ↪  Varghese', marks=95.0)
                        Student(roll='B-64', name='Divyansh
                        ↪  Aggarwal', marks=87.0)
                        Student(roll='D-65', name='Manjari Trivedi',
                        ↪  marks=44.0)
                        Student(roll='C-09', name='Kabir Sandhu',
                        ↪  marks=100.0)
                        Student(roll='C-59', name='Advik Sant',
                        ↪  marks=44.0)
                        Student(roll='F-71', name='Hazel Mani',
                        ↪  marks=48.0)
                        Student(roll='F-40', name='Adah Madan',
                        ↪  marks=57.0)
                        Student(roll='B-02', name='Ritvik Dey',
                        ↪  marks=98.0)
                        Student(roll='B-59', name='Alisha Kala',
                        ↪  marks=84.0)
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 2
    1. Create a new course
    2. View performance of all students
    3. Create course statistics
    : 3
                Course: C006
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 3
    1. Create a new batch
    2. View list of students in a batch
    3. View list of courses taught in a batch
    4. View performance of a batch
    5. Create pie chart of percentage of all students
    : 1
                Batch ID: CSE22
                Batch Name: CSE 2022-2026
                Department Name: CSE
                        Enter the courses for CSE22
                                : C002
                                : C005
                                : C008
                                : C010
                                :
                        Enter the students for CSE22
                                : CSE2221
                                : CSE2298
                                : CSE2256
                                : CSE2234
                                :
1. Student
2. Course
```

```
3. Batch
4. Department
5. Examination
: 3
    1. Create a new batch
    2. View list of students in a batch
    3. View list of courses taught in a batch
    4. View performance of a batch
    5. Create pie chart of percentage of all students
    : 2
                Batch ID: CSE96
                 ['CSE9640', 'CSE9670']
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 3
    1. Create a new batch
    2. View list of students in a batch
    3. View list of courses taught in a batch
    4. View performance of a batch
    5. Create pie chart of percentage of all students
    : 3
                Batch ID: ECE00
                          ['C001', 'C003', 'C005', 'C006', 'C007',
                           ↪  'C008', 'C009', 'C010']
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 3
    1. Create a new batch
    2. View list of students in a batch
    3. View list of courses taught in a batch
    4. View performance of a batch
    5. Create pie chart of percentage of all students
    : 4
                Batch ID: ECE02
                          Student(roll='E-31', name='Kabir Chahal',
                           ↪  percentage=69.875)
                          Student(roll='A-80', name='Arhaan Tiwari',
                           ↪  percentage=71.0)
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 3
    1. Create a new batch
    2. View list of students in a batch
    3. View list of courses taught in a batch
    4. View performance of a batch
    5. Create pie chart of percentage of all students
    : 5
                Batch ID: ECE97
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 4
    1. Create a new department
    2. View batches of a department
    3. View average performance of batches of a department
```

```
    4. Create statistics of a department
    :1
                Department ID: BA
                Department Name: Business Administration
                        Enter the batches for BA
                                : BA22
                                : BA19
                                : BA89
                                :
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 4
    1. Create a new department
    2. View batches of a department
    3. View average performance of batches of a department
    4. Create statistics of a department
    :2
                Department ID: CSE
                        ['CSE00', 'CSE05', 'CSE06', 'CSE08',
                            'CSE09', 'CSE14', 'CSE90', 'CSE92',
                         ↪  'CSE93', 'CSE95', 'CSE96', 'CSE98']
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 4
    1. Create a new department
    2. View batches of a department
    3. View average performance of batches of a department
    4. Create statistics of a department
    :3
                Department ID: IT
Performance(batch='IT03', average=73.4)
Performance(batch='IT05', average=68.6)
Performance(batch='IT08', average=67.0)
Performance(batch='IT09', average=78.1)
Performance(batch='IT11', average=72.2)
Performance(batch='IT14', average=78.7)
Performance(batch='IT91', average=74.8)
Performance(batch='IT92', average=58.0)
Performance(batch='IT96', average=61.4)
Performance(batch='IT99', average=41.3)
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 4
    1. Create a new department
    2. View batches of a department
    3. View average performance of batches of a department
    4. Create statistics of a department
    :4
                Department ID: ECE
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 5
```

```
    Hold an examination:
                        Enter the batches for exam
                                : ECE98
                                : IT92
                                : CSE09
                                :
Name of examination : Mid Semester
CSE09
        B-53
                C002: 78
                C005: 95
                C008: 48
                C010: 76
        B-22
                C002: 91
                C005: 45
                C008: 68
                C010:
ECE98
        G-73
                C001: 78
                C003: 94
                C005: 79
                C006: 48
                C007: 89
                C008: 64
                C009: 78
                C010: 97
IT92
        F-71
                C001: 74
                C003: 48
                C004: 98
                C007: 78
                C010: 46
    1. View student perfomance in the examination
    2. Create examination statistics
    : 1
[Performance(student_id='CSE0956', average=74.25),
    Performance(student_id='CSE0948', average=51.0),
    Performance(student_id='ECE9805', average=78.375),
    Performance(student_id='IT9279', average=68.8)]
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 5
    Hold an examination:
                        Enter the batches for exam
                                : ECE02
                                : ECE15
                                : IT09
                                : CSE96
                                :
Name of examination : End Semester
CSE96
        B-07
                C002: 78
                C005: 45
                C008: 68
                C010: 74
        G-50
                C002: 85
                C005: 64
                C008: 75
                C010: 48
ECE02
        E-31
                C001: 96
                C003: 45
                C005: 78
                C006: 64
                C007: 25
                C008: 69
```

```
                C009: 87
                C010: 48
        A-80
                C001: 98
                C003: 69
                C005: 78
                C006: 459
                C007: 95
                C008: 87
                C009: 79
                C010: 94
ECE15
        C-45
                C001: 91
                C003: 93
                C005: 97
                C006: 94
                C007: 75
                C008: 96
                C009: 78
                C010: 94
        D-18
                C001: 79
                C003: 85
                C005: 76
                C006: 94
                C007: 78
                C008: 94
                C009: 75
                C010: 94
IT09
        B-81
                C001: 78
                C003: 49
                C004: 96
                C007: 75
                C010: 94
        C-45
                C001: 78
                C003: 94
                C004: 78
                C007: 96
                C010: 89
    1. View student perfomance in the examination
    2. Create examination statistics
    : 2
1. Student
2. Course
3. Batch
4. Department
5. Examination
:
```

ECE1330-report_card.txt

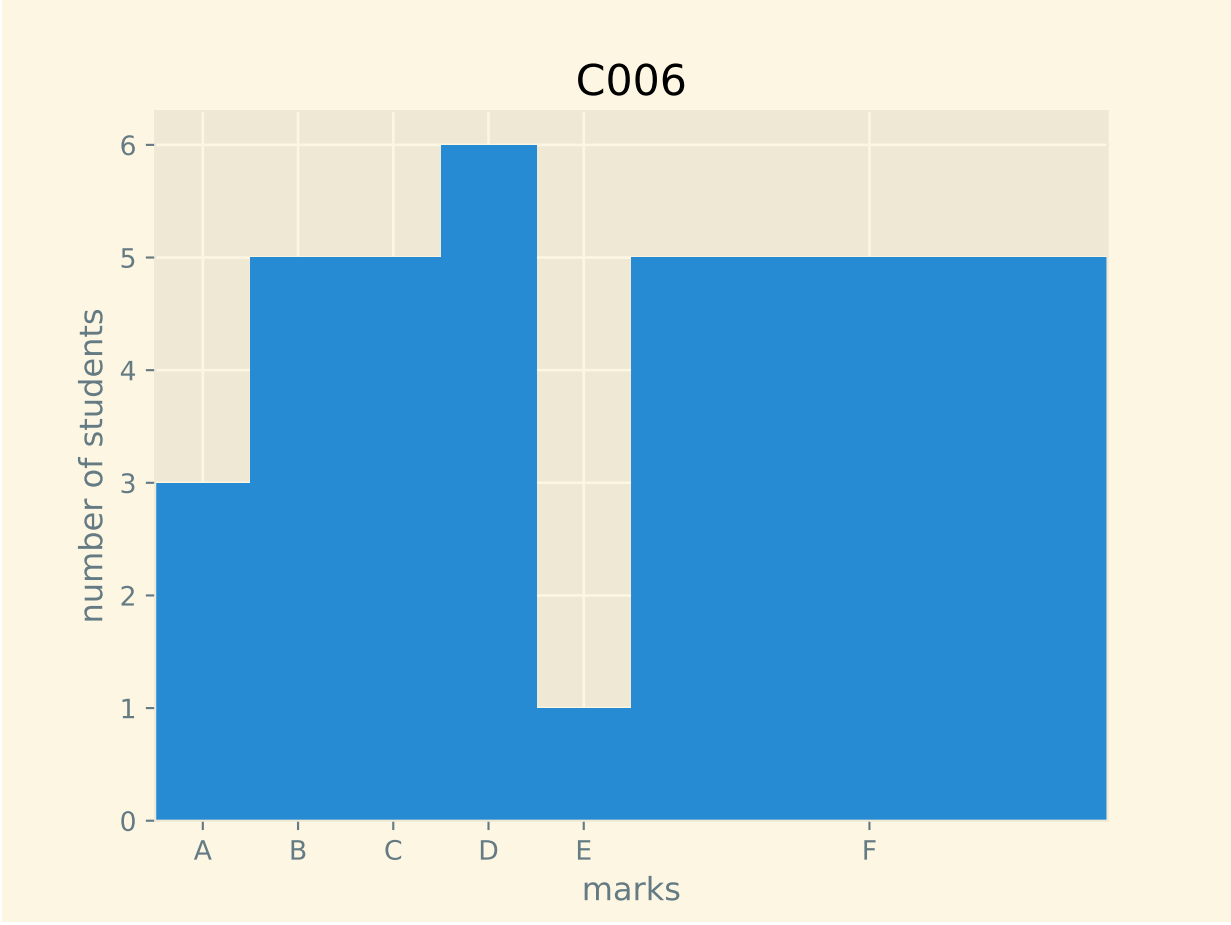
```
Sara Gade (E-95)

+------------------+-----------+----------------+------------+-------+---------+
| Course           | Course Id | Marks Obtained | Full Marks | Grade | Remarks |
+------------------+-----------+----------------+------------+-------+---------+
| Physics          | C001      |             36 |        100 |   F   | Failed  |
+------------------+-----------+----------------+------------+-------+---------+
| Biology          | C003      |             30 |        100 |   F   | Failed  |
+------------------+-----------+----------------+------------+-------+---------+
```

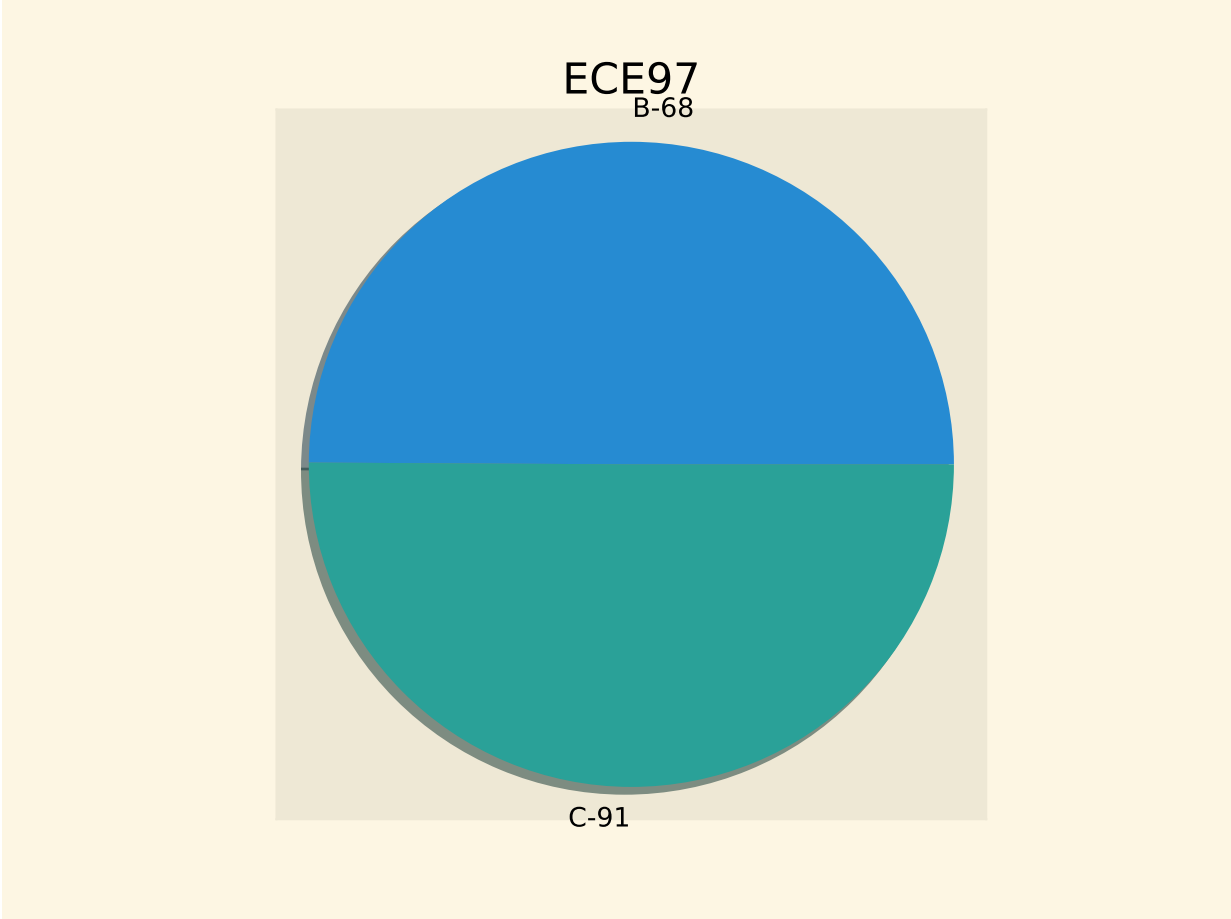
Mechanics	C005	90	100	A	Passed
Python	C006	100	100	A	Passed
Design	C007	56	100	E	Passed
Entrepreneurship	C008	46	100	F	Failed
ESP	C009	75	100	C	Passed
SDP	C010	90	100	A	Passed
Total	-	523	800	D	Passed

ID:ECE1330
Batch:ECE13

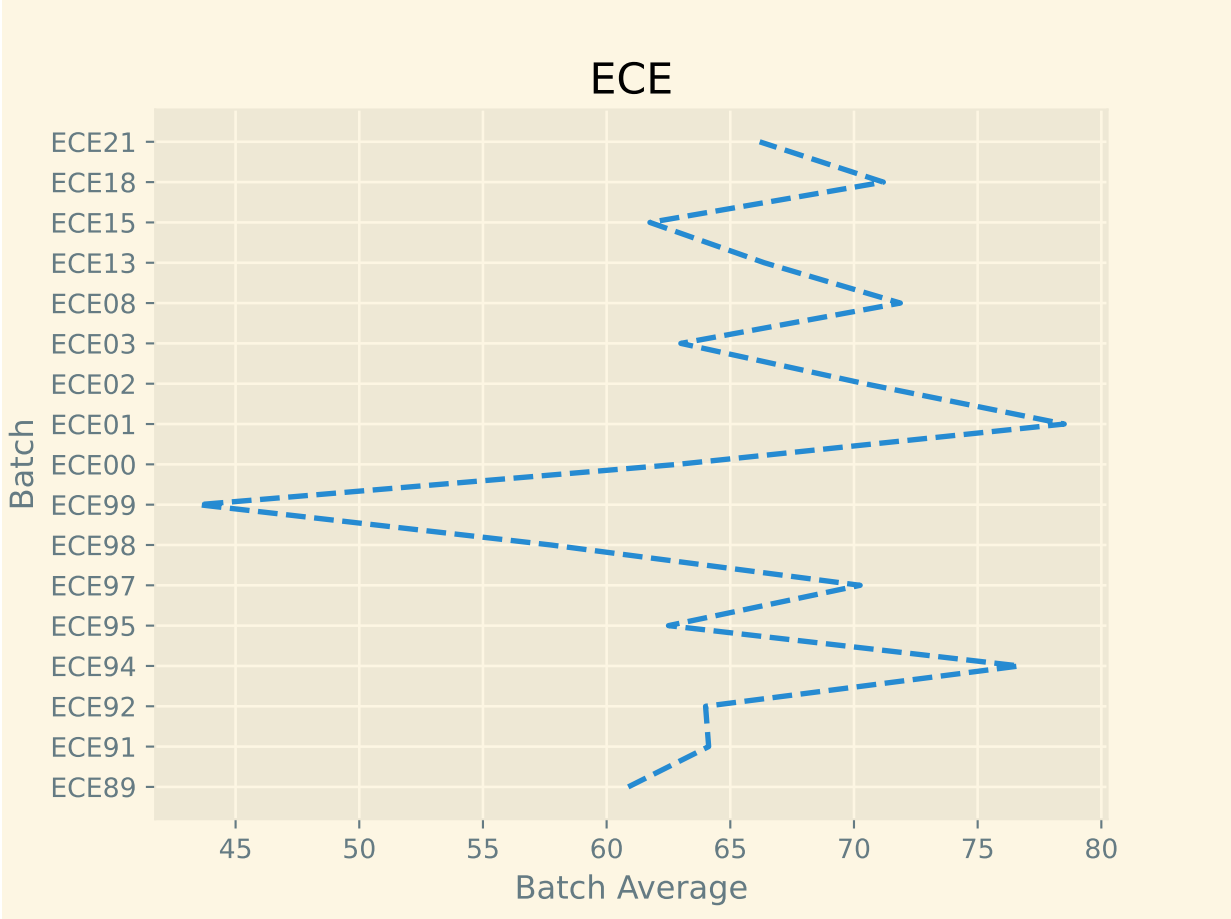
Course Statistics-C006.pdf

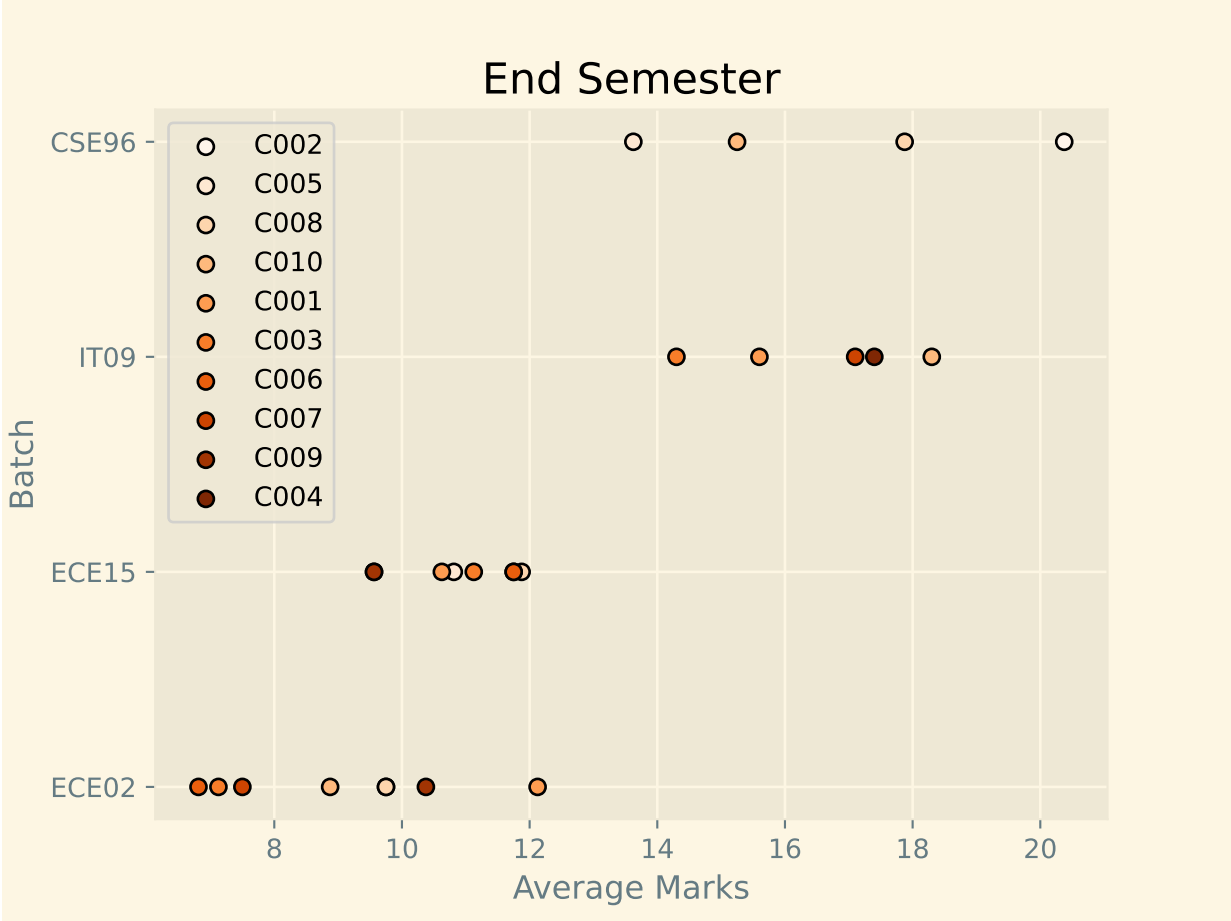


Batch Statistics-ECE97.pdf



Department Statistics-ECE.pdf




End Semester
CSE96
IT09
ECE15
ECE02
Batch
C002
C005
C008
C010
C001
C003
C006
C007
C009
C004
8
10
12
14
16
18
20
Average Marks