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
CSE01	CSE 2001-2005	CSE	…	…
CSE03	CSE 2003-2007	CSE	…	…
CSE06	CSE 2006-2010	CSE	…	…
CSE08	CSE 2008-2012	CSE	…	…
CSE11	CSE 2011-2015	CSE	…	…

CSE12	CSE 2012-2016	CSE	…	…
CSE13	CSE 2013-2017	CSE	…	…
CSE15	CSE 2015-2019	CSE	…	…
CSE17	CSE 2017-2021	CSE	…	…
CSE19	CSE 2019-2023	CSE	…	…
CSE21	CSE 2021-2025	CSE	…	…
CSE89	CSE 1989-1993	CSE	…	…
CSE90	CSE 1990-1994	CSE	…	…
CSE91	CSE 1991-1995	CSE	…	…
CSE93	CSE 1993-1997	CSE	…	…
CSE95	CSE 1995-1999	CSE	…	…
CSE97	CSE 1997-2001	CSE	…	…
CSE98	CSE 1998-2002	CSE	…	…
ECE02	ECE 2002-2006	ECE	…	…
ECE03	ECE 2003-2007	ECE	…	…
ECE05	ECE 2005-2009	ECE	…	…
ECE08	ECE 2008-2012	ECE	…	…
ECE09	ECE 2009-2013	ECE	…	…
ECE10	ECE 2010-2014	ECE	…	…
ECE13	ECE 2013-2017	ECE	…	…
ECE15	ECE 2015-2019	ECE	…	…
ECE18	ECE 2018-2022	ECE	…	…
ECE19	ECE 2019-2023	ECE	…	…
ECE20	ECE 2020-2024	ECE	…	…
ECE21	ECE 2021-2025	ECE	…	…
ECE22	ECE 2022-2026	ECE	…	…
ECE92	ECE 1992-1996	ECE	…	…
ECE94	ECE 1994-1998	ECE	…	…
ECE97	ECE 1997-2001	ECE	…	…
ECE99	ECE 1999-2003	ECE	…	…
IT00	IT 2000-2004	IT	…	…
IT05	IT 2005-2009	IT	…	…
IT07	IT 2007-2011	IT	…	…
IT08	IT 2008-2012	IT	…	…
IT09	IT 2009-2013	IT	…	…
IT12	IT 2012-2016	IT	…	…
IT15	IT 2015-2019	IT	…	…

IT19	IT 2019-2023	IT	…	…
IT90	IT 1990-1994	IT	…	…
IT91	IT 1991-1995	IT	…	…
IT92	IT 1992-1996	IT	…	…
IT93	IT 1993-1997	IT	…	…
IT94	IT 1994-1998	IT	…	…
IT96	IT 1996-2000	IT	…	…
IT99	IT 1999-2003	IT	…	…
IT22	IT 2022-2026	IT	…	…

`course.csv`

Course ID	Course Name	Marks Obtained
CSE01	CSE 2001-2005	CSE
CSE03	CSE 2003-2007	CSE
CSE06	CSE 2006-2010	CSE
CSE08	CSE 2008-2012	CSE
CSE11	CSE 2011-2015	CSE
CSE12	CSE 2012-2016	CSE
CSE13	CSE 2013-2017	CSE
CSE15	CSE 2015-2019	CSE
CSE17	CSE 2017-2021	CSE
CSE19	CSE 2019-2023	CSE
CSE21	CSE 2021-2025	CSE
CSE89	CSE 1989-1993	CSE
CSE90	CSE 1990-1994	CSE
CSE91	CSE 1991-1995	CSE
CSE93	CSE 1993-1997	CSE
CSE95	CSE 1995-1999	CSE
CSE97	CSE 1997-2001	CSE
CSE98	CSE 1998-2002	CSE
ECE02	ECE 2002-2006	ECE
ECE03	ECE 2003-2007	ECE
ECE05	ECE 2005-2009	ECE
ECE08	ECE 2008-2012	ECE
ECE09	ECE 2009-2013	ECE
ECE10	ECE 2010-2014	ECE

ECE13	ECE 2013-2017	ECE
ECE15	ECE 2015-2019	ECE
ECE18	ECE 2018-2022	ECE
ECE19	ECE 2019-2023	ECE
ECE20	ECE 2020-2024	ECE
ECE21	ECE 2021-2025	ECE
ECE22	ECE 2022-2026	ECE
ECE92	ECE 1992-1996	ECE
ECE94	ECE 1994-1998	ECE
ECE97	ECE 1997-2001	ECE
ECE99	ECE 1999-2003	ECE
IT00	IT 2000-2004	IT
IT05	IT 2005-2009	IT
IT07	IT 2007-2011	IT
IT08	IT 2008-2012	IT
IT09	IT 2009-2013	IT
IT12	IT 2012-2016	IT
IT15	IT 2015-2019	IT
IT19	IT 2019-2023	IT
IT90	IT 1990-1994	IT
IT91	IT 1991-1995	IT
IT92	IT 1992-1996	IT
IT93	IT 1993-1997	IT
IT94	IT 1994-1998	IT
IT96	IT 1996-2000	IT
IT99	IT 1999-2003	IT
IT22	IT 2022-2026	IT

`department.csv`

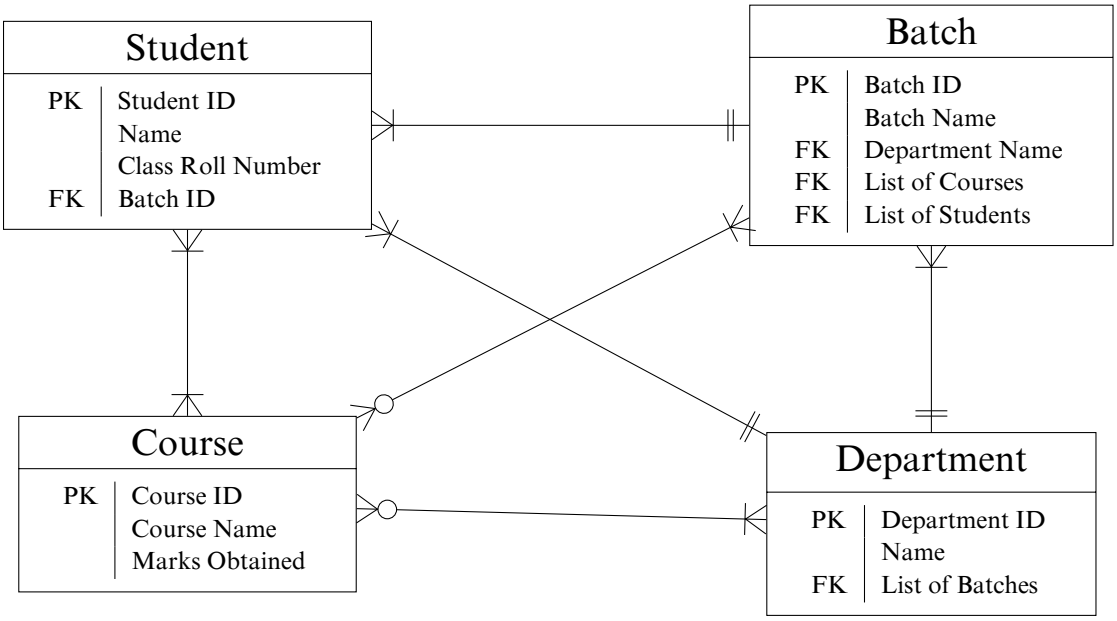
Department ID	Department Name	List of Batches
CSE	Computer Science and Engineering	…
ECE	Electronics and Communication Engineering	…
IT	Information Technology	…
BA	Business Administration	…

`student.csv`

Student ID	Name	Class Roll No	Batch ID
ECE1536	Seher Bains	B-73	ECE15
CSE0377	Rohan Konda	G-51	CSE03
ECE0893	Dharmajan Kale	E-01	ECE08
CSE9736	Mamooty Loyal	E-32	CSE97
ECE1979	Krish Bhalla	A-41	ECE19
ECE9420	Tara Rajan	D-39	ECE94
CSE1794	Ehsaan Das	C-29	CSE17
IT9321	Hazel Balan	A-04	IT93
CSE9539	Mohanlal Borra	C-06	CSE95
ECE1008	Ishaan Saraf	A-70	ECE10
IT9977	Oorja Sha	C-11	IT99
CSE1949	Miraya Kapur	D-26	CSE19
IT9911	Kiaan Mangat	H-77	IT99
ECE9429	Biju Sankaran	F-41	ECE94
CSE1521	Ryan Gokhale	H-83	CSE15
ECE9781	Lakshay Mannan	D-13	ECE97
IT0012	Vanya Kade	B-18	IT00
CSE1257	Purab Krish	G-42	CSE12
CSE2169	Alia Mann	C-97	CSE21
CSE9018	Prisha Raval	G-95	CSE90
IT0853	Kiara Lall	A-53	IT08
ECE0394	Nayantara Srinivasan	A-29	ECE03
CSE0826	Sumer Seth	E-37	CSE08
IT1959	Tanya Din	D-41	IT19
CSE8964	Dishani Tiwari	C-45	CSE89
ECE0933	Mohanlal Barman	A-06	ECE09
ECE1933	Kismat Chawla	C-33	ECE19
IT0763	Gokul Barad	B-18	IT07
CSE9849	Khushi Mangat	B-45	CSE98
ECE0807	Ishita Tank	C-77	ECE08
CSE9078	Shlok Kuruvilla	F-11	CSE90
ECE0573	Manikya Dayal	E-69	ECE05
ECE0340	Rohan Tata	A-99	ECE03
IT9186	Priyansh Bava	E-80	IT91
ECE2065	Renee Chatterjee	D-39	ECE20
CSE1343	Chirag Bobal	F-14	CSE13

CSE1391	Alisha Vaidya	F-98	CSE13
IT0994	Baiju Saraf	C-03	IT09
IT9489	Stuvan Tata	C-12	IT94
IT9269	Mannat Atwal	C-31	IT92
ECE9919	Ehsaan Toor	H-77	ECE99
IT0878	Eva Kohli	F-57	IT08
IT1243	Purab Ben	E-93	IT12
IT1519	Kashvi Chad	D-86	IT15
ECE1807	Jayan Solanki	G-62	ECE18
IT9462	Kabir Das	H-06	IT94
ECE1861	Miraan Bhatti	A-40	ECE18
CSE0691	Hrishita Sawhney	B-65	CSE06
IT9213	Lakshay Dhillon	D-42	IT92
ECE2168	Riaan Sodhi	E-57	ECE21
CSE9354	Yuvaan Sampath	A-98	CSE93
ECE1987	Mohanlal Shenoy	H-94	ECE19
IT9080	Inaaya Hegde	E-25	IT90
IT0594	Manikya Chaudry	A-65	IT05
CSE9106	Oorja Iyengar	G-72	CSE91
IT9394	Charvi Mangal	G-18	IT93
ECE9468	Ojas Sarna	C-39	ECE94
IT9634	Amani Mall	A-43	IT96
CSE0398	Stuvan Doctor	F-87	CSE03
ECE1966	Armaan Butala	A-84	ECE19
ECE2206	Eva Venkatesh	C-49	ECE22
CSE1183	Anika Sankar	G-98	CSE11
ECE0211	Tiya Balasubramanian	B-48	ECE02
CSE8939	Nitya Buch	A-59	CSE89
ECE9288	Vardaniya Thaker	D-60	ECE92
CSE9769	Zeeshan Sule	H-58	CSE97
ECE1334	Tanya Kurian	G-61	ECE13
CSE0133	Shaan Mall	H-70	CSE01
ECE0837	Prisha Rajan	C-26	ECE08
IT0010	Kartik Joshi	B-22	IT00

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
Student=namedtuple("Student",('roll','name','marks'))
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)
    marks=False
    with open('databases/course.csv','r') as csvfile:
        for row in reader(csvfile):
```

```
            if row[rown]==val and (perf:=row[2]):
                marks=perf.split('-')
                break
    if not marks:return -1
    with open('databases/student.csv','r') as csvfile:
        db=reader(csvfile)
        for perf in marks:
            student_id,mark=perf.split(':')
            for row in db:
                if row[0]==student_id:
                    yield Student(row[2],row[1],float(mark))
                    csvfile.seek(0)
                    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/ls
                             ↪  tudents
                        except KeyError:
                            try:
                                plot_data[course][batch_id]=marks
                                 ↪  /lstudents
                            except KeyError:
                                plot_data[course]={batch_id:marks
                                 ↪  /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)
        self.courses,self.data=tuple(zip(*((x,y) for x,y in
         ↪  sorted(zip(self.courses,self.data)))))#sort 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: IT0010
                Student Name: Kartik Joshi
                Class Roll No: B-22
                Batch ID: IT00
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: ECE9919
                Student Name: Ehsaan Toor
                Class Roll No: H-77
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: IT95144
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: CSE8964
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: B-65
                        Student Name: Hrishita Sawhney
                        Marks: 89
                        Class Roll Number: E-37
                        Student Name: Sumer Seth
                        Marks: 94
                        Class Roll Number: G-98
                        Student Name: Anika Sankar
                        Marks: 91
                        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='H-70', name='Shaan Mall',
                         ↪  marks=98.0)
                        Student(roll='G-51', name='Rohan Konda',
                         ↪  marks=100.0)
                        Student(roll='F-87', name='Stuvan Doctor',
                         ↪  marks=83.0)
                        Student(roll='B-65', name='Hrishita
                         ↪  Sawhney', marks=73.0)
                        Student(roll='E-37', name='Sumer Seth',
                         ↪  marks=78.0)
                        Student(roll='G-98', name='Anika Sankar',
                         ↪  marks=93.0)
                        Student(roll='G-42', name='Purab Krish',
                         ↪  marks=54.0)
                        Student(roll='F-14', name='Chirag Bobal',
                         ↪  marks=68.0)
                        Student(roll='F-98', name='Alisha Vaidya',
                         ↪  marks=83.0)
                        Student(roll='H-83', name='Ryan Gokhale',
                         ↪  marks=73.0)
                        Student(roll='C-29', name='Ehsaan Das',
                         ↪  marks=73.0)
                        Student(roll='D-26', name='Miraya Kapur',
                         ↪  marks=57.0)
                        Student(roll='C-97', name='Alia Mann',
                         ↪  marks=65.0)
```

```
Student(roll='A-59', name='Nitya Buch',
↪  marks=73.0)
Student(roll='C-45', name='Dishani Tiwari',
↪  marks=89.0)
Student(roll='G-95', name='Prisha Raval',
↪  marks=89.0)
Student(roll='F-11', name='Shlok Kuruvilla',
↪  marks=45.0)
Student(roll='G-72', name='Oorja Iyengar',
↪  marks=71.0)
Student(roll='A-98', name='Yuvaan Sampath',
↪  marks=92.0)
Student(roll='C-06', name='Mohanlal Borra',
↪  marks=66.0)
Student(roll='E-32', name='Mamooty Loyal',
↪  marks=71.0)
Student(roll='H-58', name='Zeeshan Sule',
↪  marks=65.0)
Student(roll='B-45', name='Khushi Mangat',
↪  marks=73.0)
Student(roll='B-48', name='Tiya
↪  Balasubramanian', marks=38.0)
Student(roll='A-99', name='Rohan Tata',
↪  marks=74.0)
Student(roll='A-29', name='Nayantara
↪  Srinivasan', marks=36.0)
Student(roll='E-69', name='Manikya Dayal',
↪  marks=46.0)
Student(roll='C-77', name='Ishita Tank',
↪  marks=65.0)
Student(roll='C-26', name='Prisha Rajan',
↪  marks=46.0)
Student(roll='E-01', name='Dharmajan Kale',
↪  marks=94.0)
Student(roll='A-06', name='Mohanlal Barman',
↪  marks=60.0)
Student(roll='A-70', name='Ishaan Saraf',
↪  marks=44.0)
Student(roll='G-61', name='Tanya Kurian',
↪  marks=99.0)
Student(roll='B-73', name='Seher Bains',
↪  marks=73.0)
Student(roll='G-62', name='Jayan Solanki',
↪  marks=65.0)
Student(roll='A-40', name='Miraan Bhatti',
↪  marks=95.0)
Student(roll='C-33', name='Kismat Chawla',
↪  marks=78.0)
Student(roll='A-84', name='Armaan Butala',
↪  marks=46.0)
Student(roll='A-41', name='Krish Bhalla',
↪  marks=85.0)
```

```
                        Student(roll='H-94', name='Mohanlal Shenoy',
                        ↪  marks=83.0)
                        Student(roll='D-39', name='Renee
                        ↪  Chatterjee', marks=53.0)
                        Student(roll='E-57', name='Riaan Sodhi',
                        ↪  marks=85.0)
                        Student(roll='C-49', name='Eva Venkatesh',
                        ↪  marks=61.0)
                        Student(roll='D-60', name='Vardaniya
                        ↪  Thaker', marks=71.0)
                        Student(roll='D-39', name='Tara Rajan',
                        ↪  marks=71.0)
                        Student(roll='F-41', name='Biju Sankaran',
                        ↪  marks=45.0)
                        Student(roll='C-39', name='Ojas Sarna',
                        ↪  marks=89.0)
                        Student(roll='D-13', name='Lakshay Mannan',
                        ↪  marks=77.0)
                        Student(roll='H-77', name='Ehsaan Toor',
                        ↪  marks=42.0)
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: IT22
                Batch Name: IT 2022-2026
                Department Name: IT
                        Enter the courses for IT22
                                : C001
                                : C002
                                : C003
                                : C006
                                :
                        Enter the students for IT22
                                : IT2245
                                : IT2287
                                : IT22233
                                :
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: CSE89
                 ['CSE8964', 'CSE8939']
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: CSE03
                          ['C001', 'C002', 'C005', 'C006', 'C007',
                           ↪  'C008', '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: ECE94
                          Student(roll='D-39', name='Tara Rajan',
                           ↪  percentage=74.33333333333333)
                          Student(roll='F-41', name='Biju Sankaran',
                           ↪  percentage=64.88888888888889)
                          Student(roll='C-39', name='Ojas Sarna',
                           ↪  percentage=64.44444444444444)
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: ECE19
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
                                : BA89
                                : BA13
                                :
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
                        ['CSE01', 'CSE03', 'CSE06', 'CSE08',
                            'CSE11', 'CSE12', 'CSE13', 'CSE15',
                            'CSE17', 'CSE19', 'CSE21', 'CSE89',
                            'CSE90', 'CSE91', 'CSE93', 'CSE95',
                            'CSE97', '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: CSE
Performance(batch='CSE01', average=82.0)
Performance(batch='CSE03', average=73.64285714285714)
Performance(batch='CSE06', average=78.125)
Performance(batch='CSE08', average=76.625)
Performance(batch='CSE11', average=67.875)
Performance(batch='CSE12', average=76.85714285714286)
Performance(batch='CSE13', average=73.64285714285714)
Performance(batch='CSE15', average=69.85714285714286)
Performance(batch='CSE17', average=82.85714285714286)
Performance(batch='CSE19', average=69.71428571428571)
Performance(batch='CSE21', average=70.71428571428571)
Performance(batch='CSE89', average=71.07142857142857)
Performance(batch='CSE90', average=62.92857142857143)
Performance(batch='CSE91', average=83.85714285714286)
Performance(batch='CSE93', average=74.71428571428571)
Performance(batch='CSE95', average=73.85714285714286)
Performance(batch='CSE97', average=63.785714285714285)
Performance(batch='CSE98', average=55.57142857142857)
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
                                : ECE05
                                : IT05
                                :
Name of examination : Mid Semester
ECE05
        E-69
                C001: 78
                C002: 86
                C003: 84
                C004: 96
                C005: 78
                C006: 74
                C007: 95
                C009: 96
                C010: 45
IT05
        A-65
                C001: 78
                C002: 9
                C003: 68
                C006: 74
    1. View student perfomance in the examination
    2. Create examination statistics
    : 1
[Performance(student_id='ECE0573', average=81.33333333333333),
↪  Performance(student_id='IT0594', average=57.25)]
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 5
    Hold an examination:
                        Enter the batches for exam
                                : IT08
                                : CSE08
                                : ECE08
                                :
Name of examination : End Semester
CSE08
        E-37
                C001: 78
                C002: 96
                C005: 84
                C006: 85
                C007: 75
                C008: 95
                C010: 48
                C011: 68
ECE08
        E-01
                C001: 78
                C002: 49
                C003: 56
                C004: 75
                C005: 84
                C006: 85
                C007: 96
                C009: 74
                C010: 85
        C-77
                C001: 96
                C002: 84
                C003: 75
```

```
                C004: 61
                C005: 73
                C006: 91
                C007: 73
                C009: 91
                C010: 75
        C-26
                C001: 7
                C002: 981
                C003: 73
                C004: 81
                C005: 83
                C006: 81
                C007: 73
                C009: 91
                C010: 73
IT08
        A-53
                C001: 91
                C002: 72
                C003: 82
                C006: 71
        F-57
                C001: 82
                C002: 9
                C003: 91
                C006: 92
    1. View student perfomance in the examination
    2. Create examination statistics
    : 2
1. Student
2. Course
3. Batch
4. Department
5. Examination
:
```

CSE8964-report_card.txt

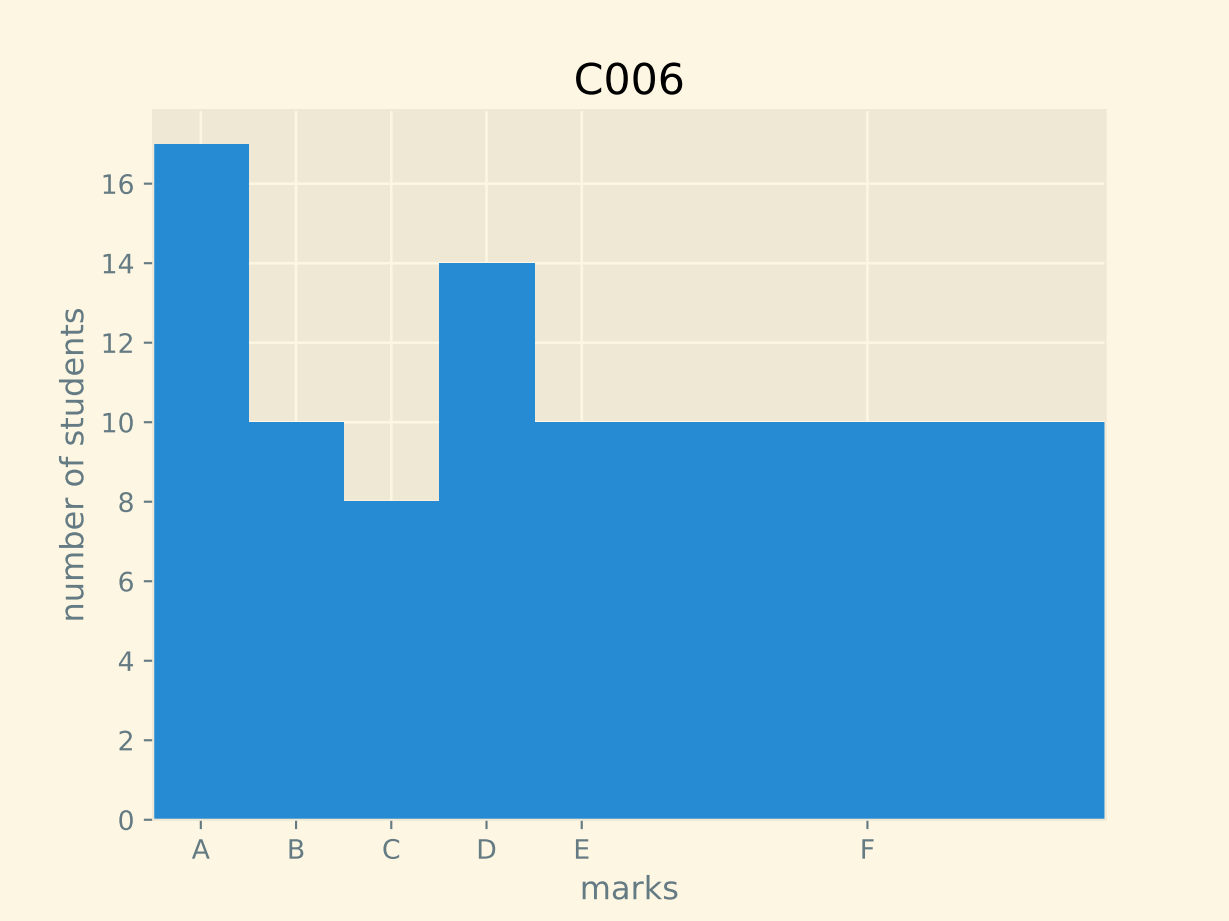
Dishani Tiwari (C-45)

Course	Course Id	Marks Obtained	Full Marks	Grade	Remarks
Physics	C001	46	100	F	Failed
Mathematics	C002	87	100	B	Passed
Mechanics	C005	84	100	B	Passed
Python	C006	85	100	B	Passed
Design	C007	90	100	A	Passed
Entrepreneurship	C008	88	100	B	Passed
SDP	C010	89	100	B	Passed

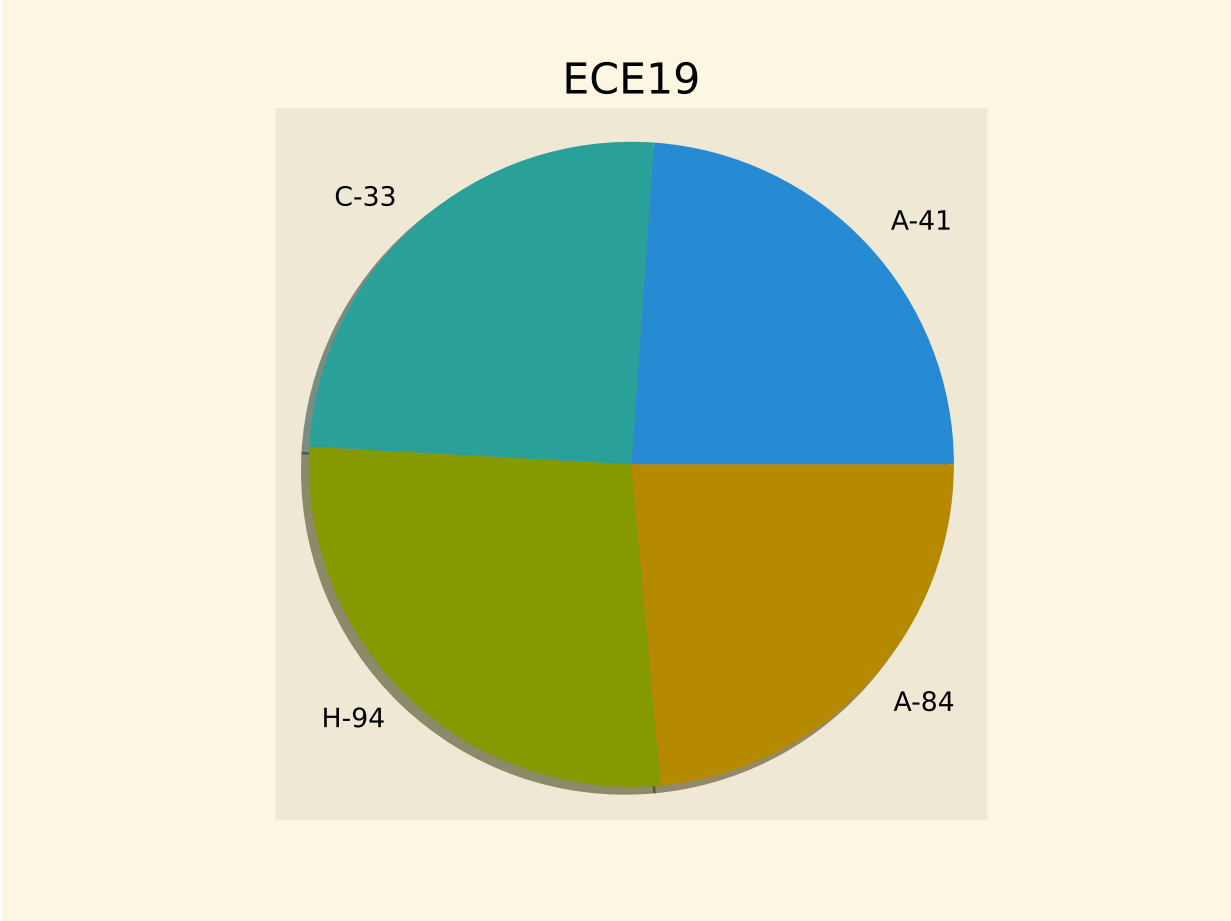
Total	-	569	700	B	Passed

ID:CSE8964
Batch:CSE89

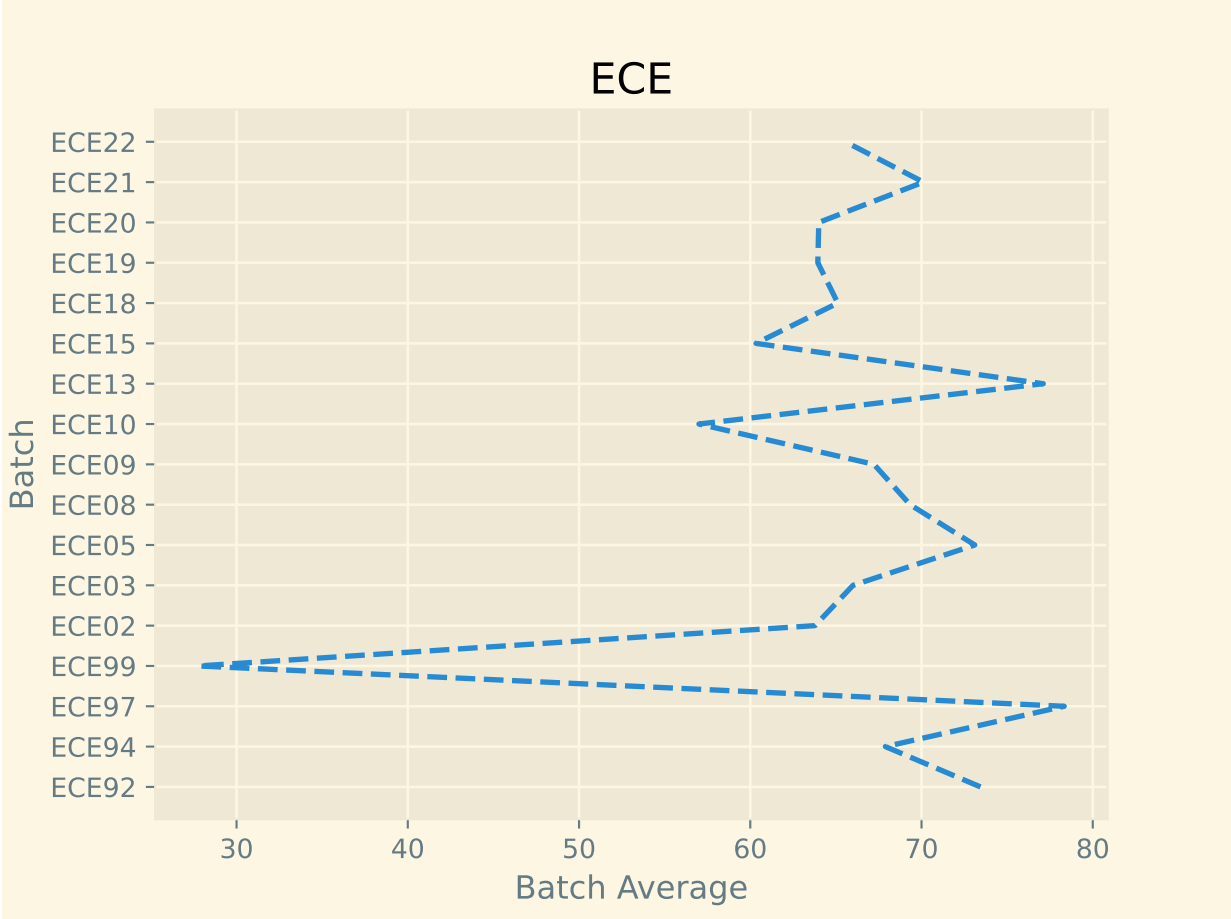
Course Statistics-C006.pdf

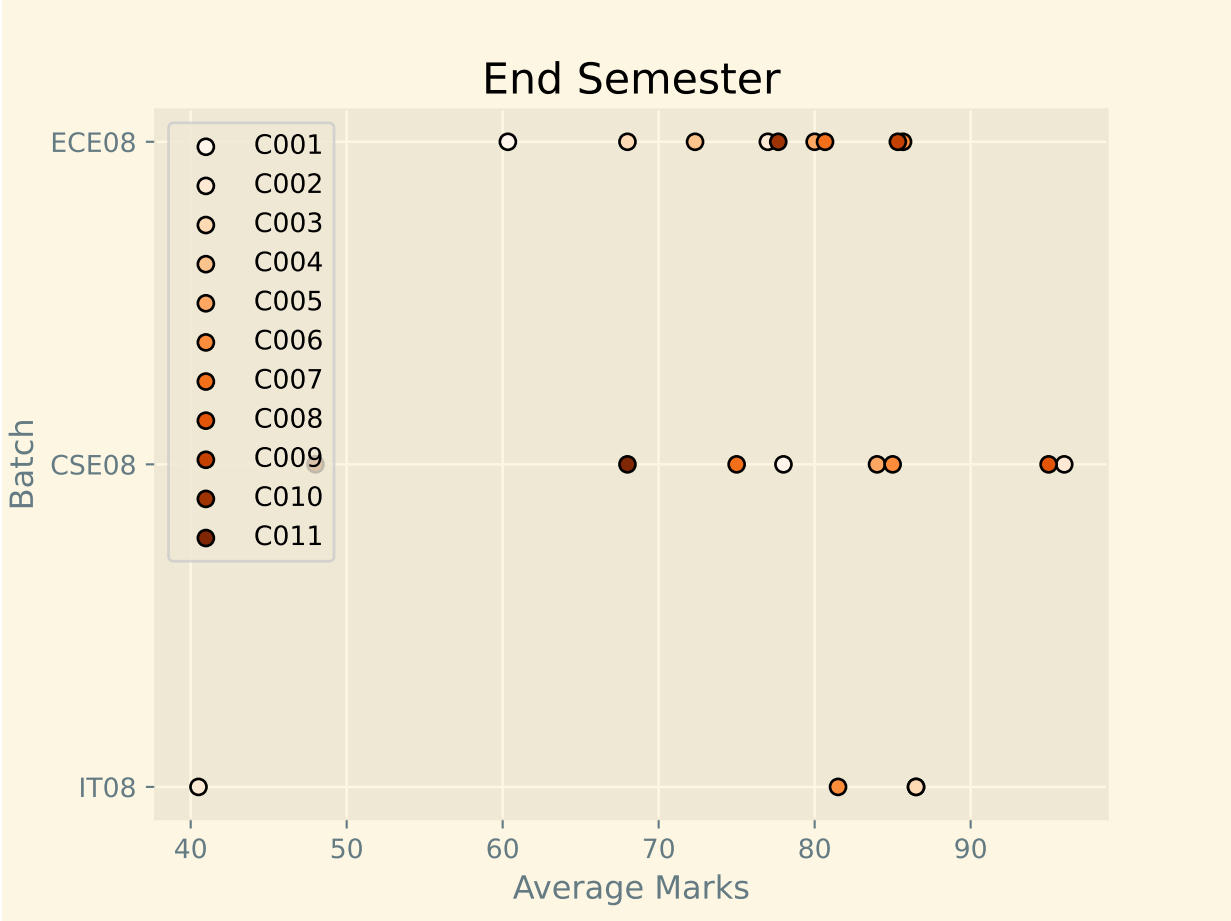


Batch Statistics-ECE19.pdf



Department Statistics-ECE.pdf




End Semester
ECE08
CSE08
IT08
Batch
C001
C002
C003
C004
C005
C006
C007
C008
C009
C010
C011
40
50
60
70
80
90
Average Marks