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

INSTITUTE OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

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	…	…
CSE01	CSE 2001-2005	CSE	…	…
CSE03	CSE 2003-2007	CSE	…	…
CSE08	CSE 2008-2012	CSE	…	…
CSE12	CSE 2012-2016	CSE	…	…

CSE13	CSE 2013-2017	CSE	…	…
CSE15	CSE 2015-2019	CSE	…	…
CSE16	CSE 2016-2020	CSE	…	…
CSE18	CSE 2018-2022	CSE	…	…
CSE19	CSE 2019-2023	CSE	…	…
CSE21	CSE 2021-2025	CSE	…	…
CSE93	CSE 1993-1997	CSE	…	…
CSE94	CSE 1994-1998	CSE	…	…
CSE96	CSE 1996-2000	CSE	…	…
CSE98	CSE 1998-2002	CSE	…	…
ECE04	ECE 2004-2008	ECE	…	…
ECE08	ECE 2008-2012	ECE	…	…
ECE10	ECE 2010-2014	ECE	…	…
ECE15	ECE 2015-2019	ECE	…	…
ECE18	ECE 2018-2022	ECE	…	…
ECE21	ECE 2021-2025	ECE	…	…
ECE92	ECE 1992-1996	ECE	…	…
ECE94	ECE 1994-1998	ECE	…	…
ECE98	ECE 1998-2002	ECE	…	…
ECE99	ECE 1999-2003	ECE	…	…
IT02	IT 2002-2006	IT	…	…
IT04	IT 2004-2008	IT	…	…
IT05	IT 2005-2009	IT	…	…
IT07	IT 2007-2011	IT	…	…
IT08	IT 2008-2012	IT	…	…
IT15	IT 2015-2019	IT	…	…
IT19	IT 2019-2023	IT	…	…
IT20	IT 2020-2024	IT	…	…
IT21	IT 2021-2025	IT	…	…
IT92	IT 1992-1996	IT	…	…
IT94	IT 1994-1998	IT	…	…
IT22	IT 2022-2026	IT	…	…

`course.csv`

Course ID	**Course Name**	**Marks Obtained**
CSE00	CSE 2000-2004	CSE

CSE01	CSE 2001-2005	CSE
CSE03	CSE 2003-2007	CSE
CSE08	CSE 2008-2012	CSE
CSE12	CSE 2012-2016	CSE
CSE13	CSE 2013-2017	CSE
CSE15	CSE 2015-2019	CSE
CSE16	CSE 2016-2020	CSE
CSE18	CSE 2018-2022	CSE
CSE19	CSE 2019-2023	CSE
CSE21	CSE 2021-2025	CSE
CSE93	CSE 1993-1997	CSE
CSE94	CSE 1994-1998	CSE
CSE96	CSE 1996-2000	CSE
CSE98	CSE 1998-2002	CSE
ECE04	ECE 2004-2008	ECE
ECE08	ECE 2008-2012	ECE
ECE10	ECE 2010-2014	ECE
ECE15	ECE 2015-2019	ECE
ECE18	ECE 2018-2022	ECE
ECE21	ECE 2021-2025	ECE
ECE92	ECE 1992-1996	ECE
ECE94	ECE 1994-1998	ECE
ECE98	ECE 1998-2002	ECE
ECE99	ECE 1999-2003	ECE
IT02	IT 2002-2006	IT
IT04	IT 2004-2008	IT
IT05	IT 2005-2009	IT
IT07	IT 2007-2011	IT
IT08	IT 2008-2012	IT
IT15	IT 2015-2019	IT
IT19	IT 2019-2023	IT
IT20	IT 2020-2024	IT
IT21	IT 2021-2025	IT
IT92	IT 1992-1996	IT
IT94	IT 1994-1998	IT
IT22	IT 2022-2026	IT

department.csv

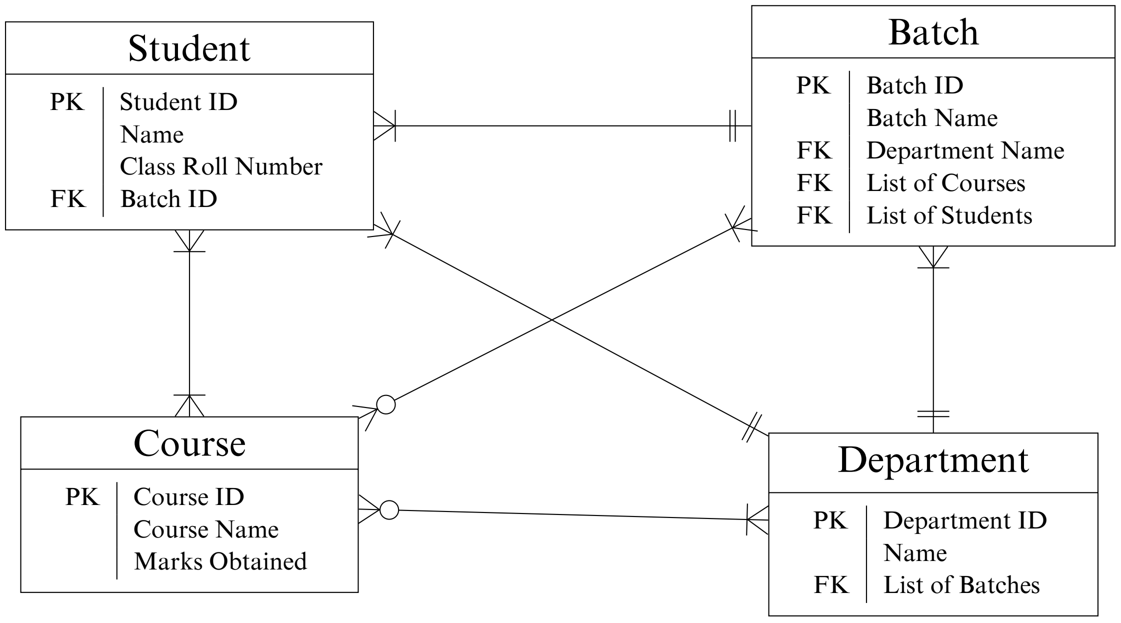
Department ID	Department Name	List of Batches
CSE	Computer Science and Engineering	…
ECE	Electronics and Communication Engineering	…
IT	Information Technology	…
BA	Business Administration	…

student.csv

Student ID	Name	Class Roll No	Batch ID
CSE0388	Shalv Warrior	E-06	CSE03
IT2119	Jayan Dugal	E-18	IT21
IT1594	Tejas Kari	H-95	IT15
CSE1314	Ahana Chakraborty	C-44	CSE13
ECE9442	Kashvi Saha	G-66	ECE94
CSE1331	Anahita Tank	C-70	CSE13
CSE1213	Advika Aurora	C-68	CSE12
IT9265	Renee Dube	D-90	IT92
CSE1579	Diya Sane	E-57	CSE15
ECE0871	Onkar Krish	E-92	ECE08
ECE9805	Oorja Trivedi	B-95	ECE98
IT9209	Hazel Biswas	B-94	IT92
ECE9264	Hrishita Sura	F-41	ECE92
IT9411	Sana Batta	B-63	IT94
CSE9862	Emir Tella	D-80	CSE98
ECE9906	Indranil Shah	E-18	ECE99
ECE9219	Sana Sahota	C-83	ECE92
ECE0444	Vihaan Wali	B-19	ECE04
CSE1664	Stuvan Iyengar	F-97	CSE16
CSE0145	Shlok Behl	C-63	CSE01
CSE0062	Nirvi Deshpande	E-08	CSE00
ECE1856	Anahita Korpal	C-26	ECE18
ECE9841	Anya Kale	H-83	ECE98
CSE1980	Mehul Bahri	G-51	CSE19
CSE9350	Ela Kashyap	D-55	CSE93
CSE9370	Tushar Vasa	F-59	CSE93
ECE9932	Riya Dasgupta	C-26	ECE99

IT0226	Nirvaan Atwal	G-13	IT02
ECE9280	Hridaan Sawhney	G-08	ECE92
ECE1042	Nehmat Shetty	G-77	ECE10
CSE2139	Gatik Dara	B-94	CSE21
ECE2149	Aarna Gandhi	C-45	ECE21
IT2121	Yuvraj Borra	F-40	IT21
ECE9961	Renee Devan	B-63	ECE99
CSE0878	Armaan Venkatesh	E-07	CSE08
IT1929	Neelofar Wason	H-24	IT19
CSE9618	Hiran Salvi	F-66	CSE96
ECE9857	Azad Dada	C-65	ECE98
IT0707	Nirvaan Jhaveri	F-11	IT07
IT0843	Drishya Bhat	A-25	IT08
CSE1205	Romil Maharaj	B-94	CSE12
IT0477	Vidur Bandi	F-94	IT04
ECE1856	Ryan Shere	E-31	ECE18
CSE9861	Hridaan Swamy	H-94	CSE98
ECE9279	Ehsaan Rastogi	G-35	ECE92
ECE9262	Raghav Subramanian	C-53	ECE92
CSE9460	Yakshit Venkataraman	H-12	CSE94
IT0566	Mahika Karan	C-71	IT05
CSE1824	Onkar Chokshi	G-35	CSE18
ECE2180	Faiyaz Kumar	E-51	ECE21
IT0242	Jhanvi Dar	A-18	IT02
ECE9484	Zoya Krishna	H-18	ECE94
IT2093	Amani Ravel	B-83	IT20
ECE1563	Azad Vaidya	G-94	ECE15
ECE1034	Veer Chauhan	D-02	ECE10
CSE1853	Himmat Salvi	B-12	CSE18
ECE1564	Mohanlal Mand	A-80	ECE15
CSE1239	Nayantara Vyas	H-43	CSE12
IT0234	Kartik Joshi	B-22	IT02

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()
```

department.py

```
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: IT0234
                Student Name: Kartik Joshi
                Class Roll No: B-22
                Batch ID: IT02
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: CSE0062
                Student Name: Nirvi Deshpande
                Class Roll No: E-08
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: ECE1666
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: CSE1314
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: C-63
                        Student Name: Shlok Behl
                        Marks: 89
                        Class Roll Number: E-06
                        Student Name: Shalv Warrior
                        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='E-08', name='Nirvi Deshpande',
                          ↪  marks=94.0)
                         Student(roll='C-63', name='Shlok Behl',
                          ↪  marks=85.0)
                         Student(roll='E-06', name='Shalv Warrior',
                          ↪  marks=30.0)
                         Student(roll='E-07', name='Armaan
                          ↪  Venkatesh', marks=30.0)
                         Student(roll='B-94', name='Romil Maharaj',
                          ↪  marks=34.0)
                         Student(roll='C-68', name='Advika Aurora',
                          ↪  marks=36.0)
                         Student(roll='H-43', name='Nayantara Vyas',
                          ↪  marks=74.0)
                         Student(roll='C-44', name='Ahana
                          ↪  Chakraborty', marks=84.0)
                         Student(roll='C-70', name='Anahita Tank',
                          ↪  marks=72.0)
                         Student(roll='E-57', name='Diya Sane',
                          ↪  marks=44.0)
                         Student(roll='F-97', name='Stuvan Iyengar',
                          ↪  marks=57.0)
                         Student(roll='G-35', name='Onkar Chokshi',
                          ↪  marks=91.0)
                         Student(roll='B-12', name='Himmat Salvi',
                          ↪  marks=100.0)
                         Student(roll='G-51', name='Mehul Bahri',
                          ↪  marks=97.0)
```

```
                              Student(roll='B-94', name='Gatik Dara',
                               ↪  marks=65.0)
                              Student(roll='D-55', name='Ela Kashyap',
                               ↪  marks=67.0)
                              Student(roll='F-59', name='Tushar Vasa',
                               ↪  marks=96.0)
                              Student(roll='H-12', name='Yakshit
                               ↪  Venkataraman', marks=91.0)
                              Student(roll='F-66', name='Hiran Salvi',
                               ↪  marks=42.0)
                              Student(roll='H-94', name='Hridaan Swamy',
                               ↪  marks=70.0)
                              Student(roll='D-80', name='Emir Tella',
                               ↪  marks=52.0)
                              Student(roll='G-13', name='Nirvaan Atwal',
                               ↪  marks=89.0)
                              Student(roll='A-18', name='Jhanvi Dar',
                               ↪  marks=42.0)
                              Student(roll='F-94', name='Vidur Bandi',
                               ↪  marks=69.0)
                              Student(roll='C-71', name='Mahika Karan',
                               ↪  marks=77.0)
                              Student(roll='F-11', name='Nirvaan Jhaveri',
                               ↪  marks=39.0)
                              Student(roll='A-25', name='Drishya Bhat',
                               ↪  marks=79.0)
                              Student(roll='H-95', name='Tejas Kari',
                               ↪  marks=99.0)
                              Student(roll='H-24', name='Neelofar Wason',
                               ↪  marks=80.0)
                              Student(roll='B-83', name='Amani Ravel',
                               ↪  marks=57.0)
                              Student(roll='E-18', name='Jayan Dugal',
                               ↪  marks=85.0)
                              Student(roll='F-40', name='Yuvraj  Borra',
                               ↪  marks=51.0)
                              Student(roll='B-94', name='Hazel Biswas',
                               ↪  marks=100.0)
                              Student(roll='D-90', name='Renee Dube',
                               ↪  marks=79.0)
                              Student(roll='B-63', name='Sana Batta',
                               ↪  marks=85.0)
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
```

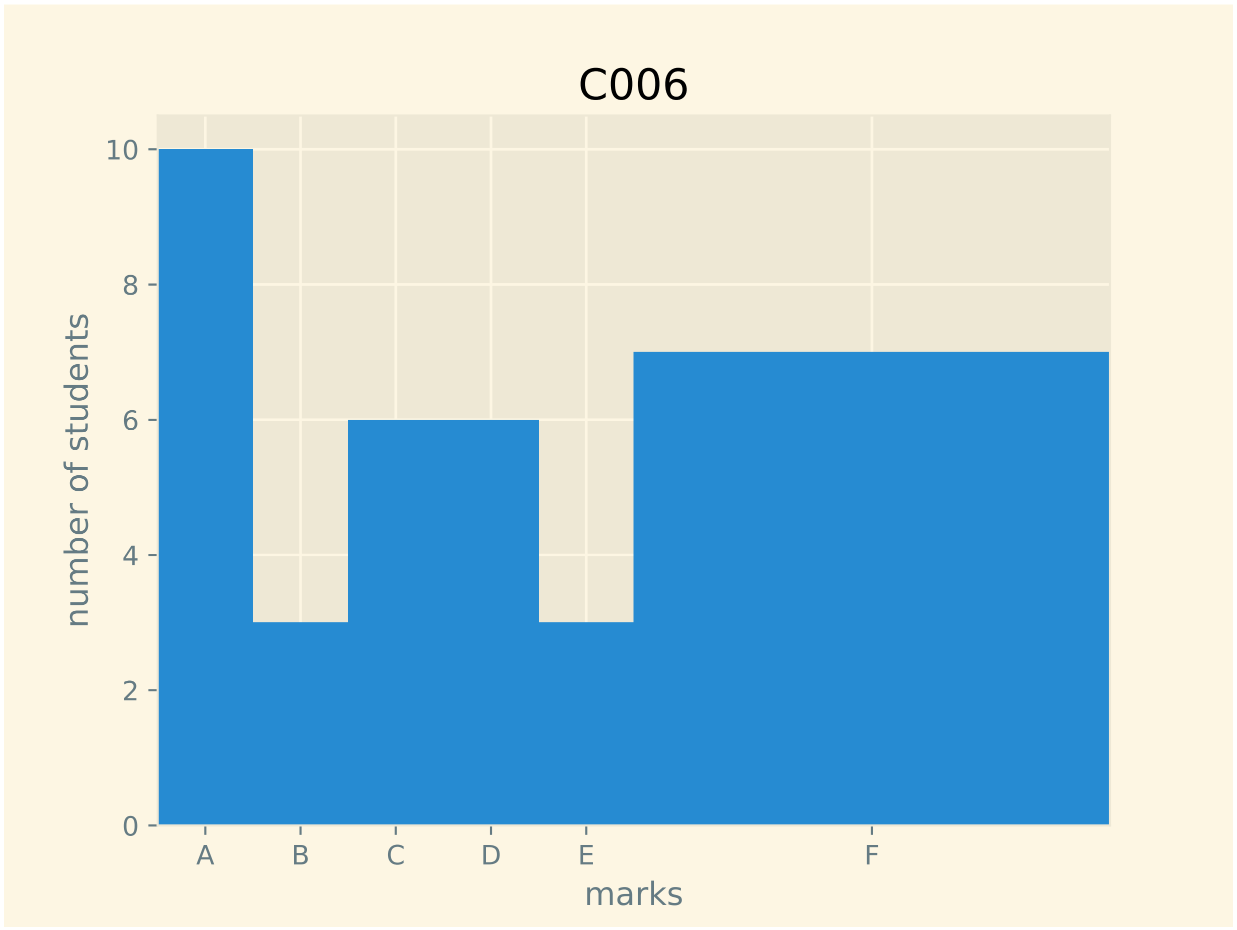
```
5. Examination
: 3
    1. Create a new batch
    2. View list of students in a batch
    3. View list of courses taught in a batch
    4. View performance of a batch
    5. Create pie chart of percentage of all students
    : 1
                Batch ID: IT22
                Batch Name: IT 2022-2026
                Department Name: IT
                        Enter the courses for IT22
                                : C003
                                : C004
                                : C005
                                : C006
                                : C009
                                : C010
                                :
                        Enter the students for IT22
                                : IT2278
                                : IT2256
                                : IT2233
                                :
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: IT21
                 ['IT2119', 'IT2121']
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: CSE12
                         Student(roll='C-68', name='Advika Aurora',
                          ↪  percentage=68.8)
                         Student(roll='B-94', name='Romil Maharaj',
                          ↪  percentage=67.1)
                         Student(roll='H-43', name='Nayantara Vyas',
                          ↪  percentage=70.2)
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: ECE92
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
                               : BA14
                               :
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', 'CSE01', 'CSE03', 'CSE08',
                             'CSE12', 'CSE13', 'CSE15', 'CSE16',
                             'CSE18', 'CSE19', 'CSE21', 'CSE93',
                        ↪↪↪  'CSE94', '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: ECE
Performance(batch='ECE04', average=81.0)
Performance(batch='ECE08', average=80.0)
Performance(batch='ECE10', average=47.5)
Performance(batch='ECE15', average=39.0)
Performance(batch='ECE18', average=39.0)
Performance(batch='ECE21', average=48.0)
Performance(batch='ECE92', average=73.4)
Performance(batch='ECE94', average=56.0)
Performance(batch='ECE98', average=78.0)
Performance(batch='ECE99', average=85.0)
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: CSE
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 5
    Hold an examination:
                        Enter the batches for exam
                               : CSE15
                               : IT15
                               : ECE15
                               :
Name of examination : MideSemester
CSE15
        E-57
                C001: 78
                C002: 96
                C003: 75
                C004: 74
                C005: 53
                C006: 6
                C007: 48
                C008: 97
                C009: 67
                C010: 88
ECE15
        G-94
                C009: 96
        A-80
                C009: 74
IT15
        H-95
                C003: 48
                C004: 56
                C005: 89
                C006: 74
                C009: 89
                C010: 75
    1. View student perfomance in the examination
    2. Create examination statistics
    : 1
[Performance(student_id='CSE1579', average=68.2),
    Performance(student_id='ECE1563', average=96.0),
    Performance(student_id='ECE1564', average=74.0),
    Performance(student_id='IT1594', average=71.83333333333333)]
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 5
    Hold an examination:
                        Enter the batches for exam
                               : CSE18
                               : ECE18
                               : IT18
                               :
Name of examination : End Semester
CSE18
        G-35
                C001: 89
                C002: 87
                C003: 86
                C004: 74
                C005: 85
                C006: 96
                C007: 78
                C008: 74
                C009: 89
                C010: 65
        B-12
                C001: 78
                C002: 74
                C003: 89
                C004: 85
                C005: 96
                C006: 78
                C007: 74
                C008: 85
                C009: 89
                C010: 86
ECE18
        C-26
                C009: 87
        C-26
                C009: 74
    1. View student perfomance in the examination
    2. Create examination statistics
    : 2
1. Student
2. Course
3. Batch
4. Department
5. Examination
:
```

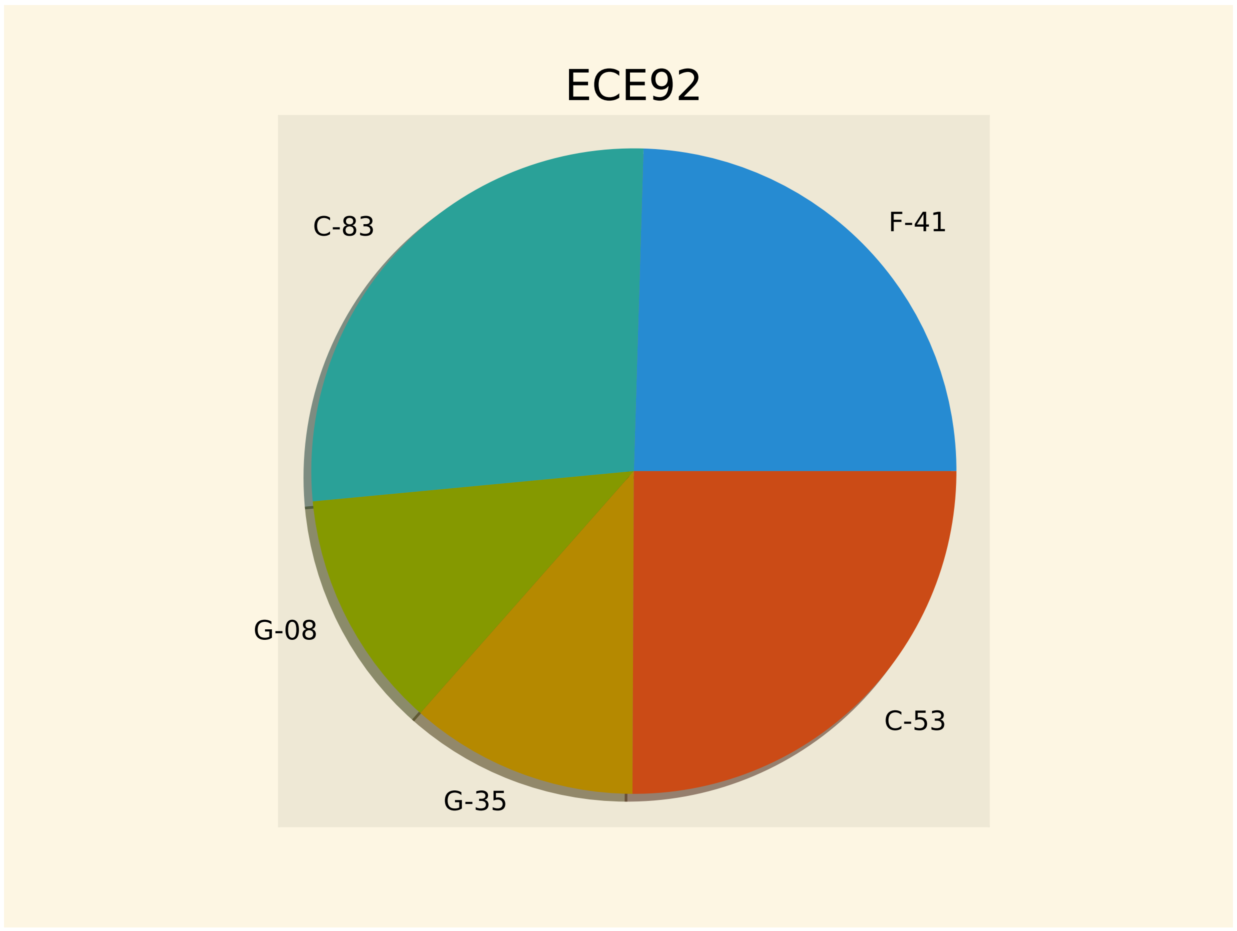
CSE1314-report_card.txt

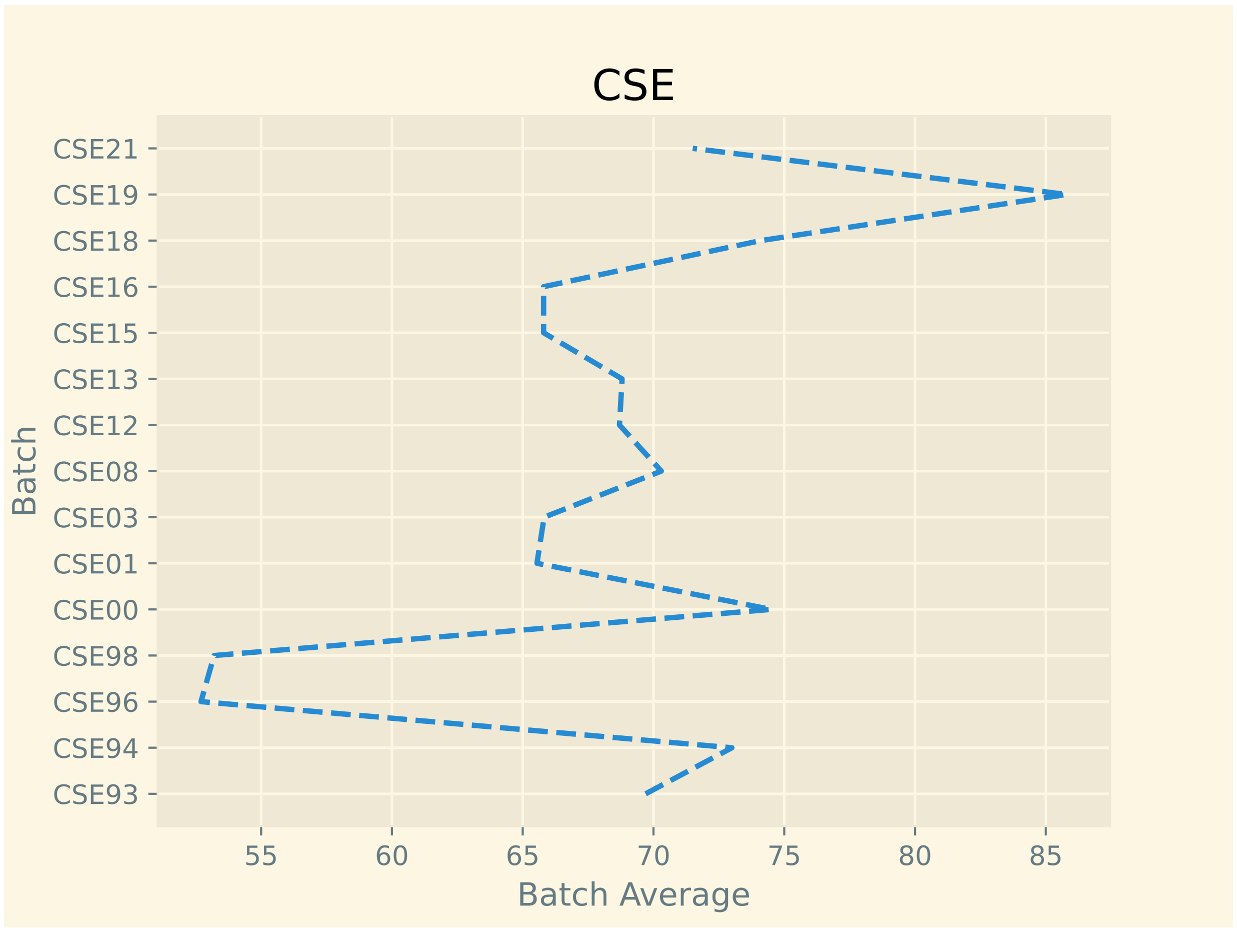
```
Ahana  Chakraborty (C-44)

+------------------+-----------+----------------+------------+-------+---------+
| Course           | Course Id | Marks Obtained | Full Marks | Grade | Remarks |
+------------------+-----------+----------------+------------+-------+---------+
| Physics          | C001      |             94 |        100 |   A   | Passed  |
+------------------+-----------+----------------+------------+-------+---------+
| Mathematics      | C002      |             47 |        100 |   F   | Failed  |
+------------------+-----------+----------------+------------+-------+---------+
| Biology          | C003      |             87 |        100 |   B   | Passed  |
+------------------+-----------+----------------+------------+-------+---------+
| Electrical       | C004      |             53 |        100 |   E   | Passed  |
+------------------+-----------+----------------+------------+-------+---------+
| Mechanics        | C005      |             46 |        100 |   F   | Failed  |
+------------------+-----------+----------------+------------+-------+---------+
| Python           | C006      |             91 |        100 |   A   | Passed  |
+------------------+-----------+----------------+------------+-------+---------+
| Design           | C007      |             94 |        100 |   A   | Passed  |
+------------------+-----------+----------------+------------+-------+---------+
| Entrepreneurship | C008      |             38 |        100 |   F   | Failed  |
+------------------+-----------+----------------+------------+-------+---------+
| ESP              | C009      |             73 |        100 |   C   | Passed  |
+------------------+-----------+---------------+------------+-------+---------+
| SDP              | C010      |            84 |        100 |   B   | Passed  |
+------------------+-----------+---------------+------------+-------+---------+
| Total            | -         |           707 |       1000 |   C   | Passed  |
+------------------+-----------+---------------+------------+-------+---------+

ID:CSE1314
Batch:CSE13
```

Course Statistics-C006.pdf

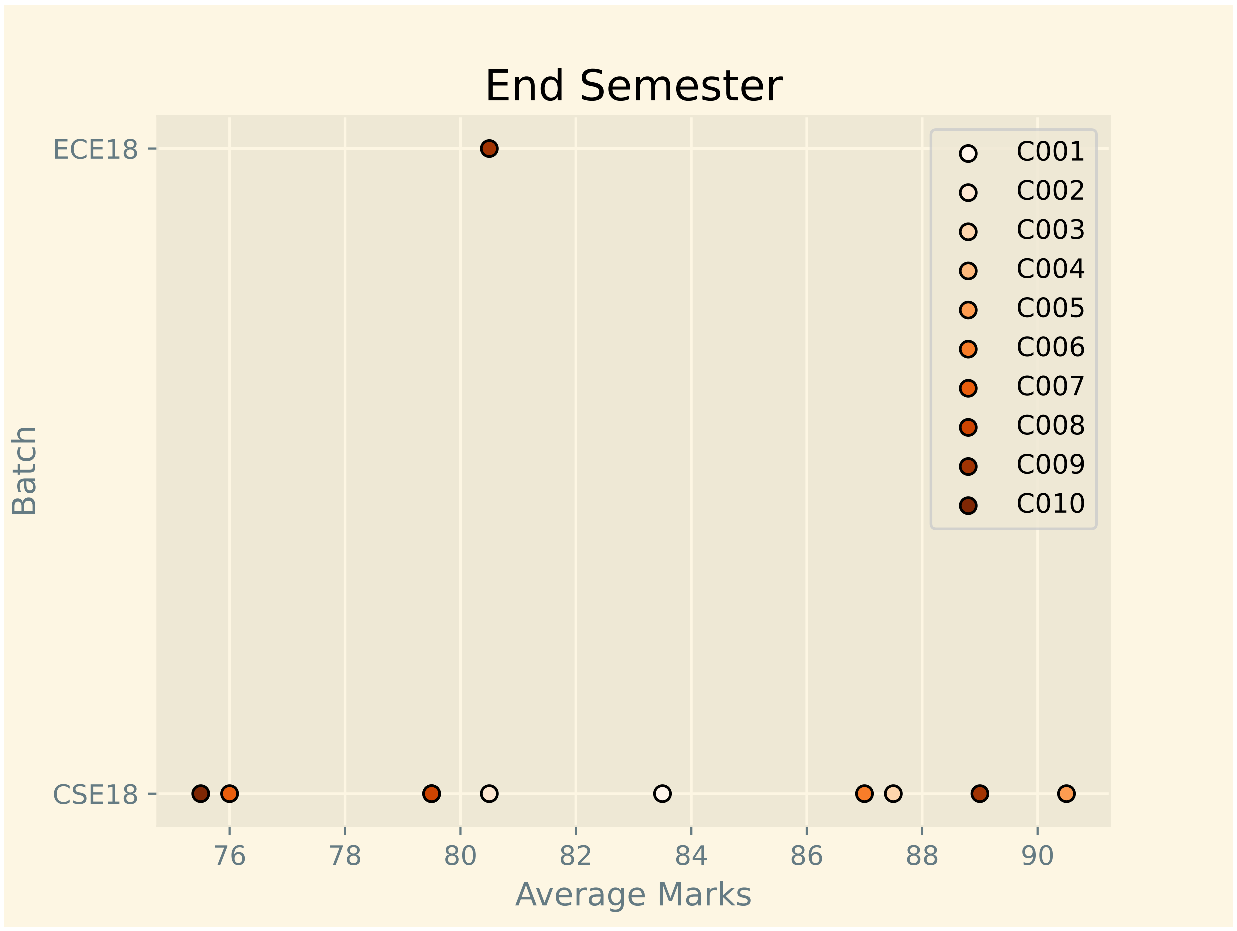


Batch Statistics-ECE92.pdf




CSE
Batch
CSE21
CSE19
CSE18
CSE16
CSE15
CSE13
CSE12
CSE08
CSE03
CSE01
CSE00
CSE98
CSE96
CSE94
CSE93
55
60
65
70
75
80
85
Batch Average

End Semester Exam.pdf


End Semester
Batch
ECE18
CSE18
Average Marks
C001
C002
C003
C004
C005
C006
C007
C008
C009
C010