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
CSE05	CSE 2005-2009	CSE	…	…
CSE07	CSE 2007-2011	CSE	…	…
CSE11	CSE 2011-2015	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	…	…
CSE89	CSE 1989-1993	CSE	…	…
CSE90	CSE 1990-1994	CSE	…	…
CSE91	CSE 1991-1995	CSE	…	…
CSE92	CSE 1992-1996	CSE	…	…
CSE94	CSE 1994-1998	CSE	…	…
CSE95	CSE 1995-1999	CSE	…	…
CSE96	CSE 1996-2000	CSE	…	…
CSE98	CSE 1998-2002	CSE	…	…
ECE00	ECE 2000-2004	ECE	…	…
ECE02	ECE 2002-2006	ECE	…	…
ECE03	ECE 2003-2007	ECE	…	…
ECE04	ECE 2004-2008	ECE	…	…
ECE06	ECE 2006-2010	ECE	…	…
ECE08	ECE 2008-2012	ECE	…	…
ECE09	ECE 2009-2013	ECE	…	…
ECE10	ECE 2010-2014	ECE	…	…
ECE12	ECE 2012-2016	ECE	…	…
ECE13	ECE 2013-2017	ECE	…	…
ECE14	ECE 2014-2018	ECE	…	…
ECE22	ECE 2022-2026	ECE	…	…
ECE90	ECE 1990-1994	ECE	…	…
ECE91	ECE 1991-1995	ECE	…	…
ECE92	ECE 1992-1996	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	…	…
IT16	IT 2016-2020	IT	…	…
IT17	IT 2017-2021	IT	…	…
IT21	IT 2021-2025	IT	…	…

IT22	IT 2022-2026	IT	…	…
IT91	IT 1991-1995	IT	…	…
IT92	IT 1992-1996	IT	…	…
IT95	IT 1995-1999	IT	…	…
IT96	IT 1996-2000	IT	…	…
IT97	IT 1997-2001	IT	…	…
IT20	IT 2000-2024	IT	…	…

`course.csv`

Course ID	Course Name	Marks Obtained
CSE05	CSE 2005-2009	CSE
CSE07	CSE 2007-2011	CSE
CSE11	CSE 2011-2015	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
CSE89	CSE 1989-1993	CSE
CSE90	CSE 1990-1994	CSE
CSE91	CSE 1991-1995	CSE
CSE92	CSE 1992-1996	CSE
CSE94	CSE 1994-1998	CSE
CSE95	CSE 1995-1999	CSE
CSE96	CSE 1996-2000	CSE
CSE98	CSE 1998-2002	CSE
ECE00	ECE 2000-2004	ECE
ECE02	ECE 2002-2006	ECE
ECE03	ECE 2003-2007	ECE
ECE04	ECE 2004-2008	ECE
ECE06	ECE 2006-2010	ECE
ECE08	ECE 2008-2012	ECE
ECE09	ECE 2009-2013	ECE
ECE10	ECE 2010-2014	ECE

ECE12	ECE 2012-2016	ECE
ECE13	ECE 2013-2017	ECE
ECE14	ECE 2014-2018	ECE
ECE22	ECE 2022-2026	ECE
ECE90	ECE 1990-1994	ECE
ECE91	ECE 1991-1995	ECE
ECE92	ECE 1992-1996	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
IT16	IT 2016-2020	IT
IT17	IT 2017-2021	IT
IT21	IT 2021-2025	IT
IT22	IT 2022-2026	IT
IT91	IT 1991-1995	IT
IT92	IT 1992-1996	IT
IT95	IT 1995-1999	IT
IT96	IT 1996-2000	IT
IT97	IT 1997-2001	IT
IT20	IT 2000-2024	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	…

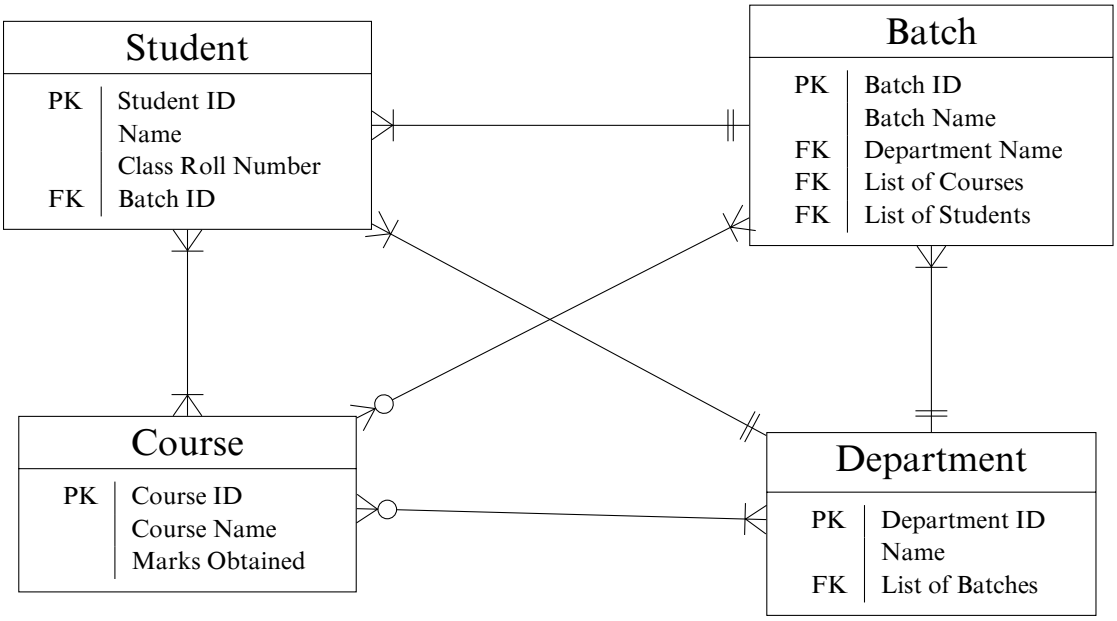
`student.csv`

Student ID	Name	Class Roll No	Batch ID
IT9675	Kiaan Sankaran	E-82	IT96
IT9179	Veer Chaudhari	D-18	IT91

IT0851	Yashvi Sandal	B-71	IT08
IT1579	Jiya Ratti	F-33	IT15
IT9262	Shalv Kaur	E-68	IT92
CSE1988	Damini Bhakta	A-94	CSE19
CSE9639	Rania Date	F-78	CSE96
ECE9135	Dhanush Jhaveri	G-43	ECE91
CSE9153	Lagan Bahl	A-28	CSE91
CSE1548	Keya Dubey	C-80	CSE15
CSE8906	Hiran Sabharwal	H-80	CSE89
ECE1494	Adah Contractor	C-14	ECE14
CSE1566	Tara Sahni	F-39	CSE15
IT9718	Akarsh Vyas	E-94	IT97
ECE0231	Khushi Chatterjee	G-49	ECE02
IT9580	Pihu Agrawal	D-53	IT95
IT0235	Raghav Sachar	D-42	IT02
IT1748	Lagan Ahluwalia	H-80	IT17
ECE0228	Nitya Kari	C-21	ECE02
ECE0059	Elakshi Chaudhry	H-06	ECE00
CSE8941	Inaaya Bhakta	C-19	CSE89
CSE9030	Ryan Kata	H-36	CSE90
CSE1335	Shanaya Doctor	B-39	CSE13
ECE0845	Advik Kaur	F-06	ECE08
ECE1339	Taimur Dhaliwal	C-98	ECE13
IT0851	Devansh Bail	C-89	IT08
CSE9853	Fateh Raval	B-80	CSE98
ECE0339	Nehmat Dhillon	H-62	ECE03
IT9766	Adira Sengupta	E-40	IT97
CSE9177	Akarsh Sunder	H-89	CSE91
CSE1939	Emir Rajagopal	D-44	CSE19
IT0479	Dharmajan Johal	G-01	IT04
ECE0956	Sumer Kalita	G-48	ECE09
IT9299	Chirag Ramanathan	C-66	IT92
IT9258	Uthkarsh Sen	A-42	IT92
ECE2206	Yasmin Baral	H-45	ECE22
CSE0519	Baiju Kala	B-48	CSE05
CSE1332	Jayan Krish	A-51	CSE13
IT1645	Kavya Das	H-56	IT16

CSE1939	Kiara Sundaram	A-18	CSE19
ECE1059	Hrishita Kunda	A-65	ECE10
IT0783	Zara Kunda	H-54	IT07
CSE1681	Vedika Uppal	C-54	CSE16
CSE1878	Advika Kapur	C-59	CSE18
CSE9135	Yakshit Dixit	B-59	CSE91
ECE0649	Taimur Kibe	G-18	ECE06
CSE9643	Anika Dhawan	F-33	CSE96
IT2284	Saanvi Divan	G-07	IT22
CSE9533	Umang De	B-04	CSE95
ECE9098	Sana Mann	H-66	ECE90
CSE0573	Anvi Banerjee	D-20	CSE05
IT1559	Amira Date	H-71	IT15
CSE2191	Nitya Chhabra	F-98	CSE21
CSE9686	Misha Vala	F-88	CSE96
IT2114	Zeeshan Bali	B-55	IT21
CSE1135	Anahita Kota	C-42	CSE11
ECE1213	Vardaniya Subramanian	H-89	ECE12
CSE9472	Damini Chahal	D-26	CSE94
ECE0278	Dhanuk Deol	D-49	ECE02
ECE9256	Ahana Agarwal	A-56	ECE92
ECE0406	Amani Handa	E-66	ECE04
CSE0780	Vanya Suri	H-55	CSE07
CSE1271	Shlok Bhavsar	E-50	CSE12
CSE9158	Vihaan Sarraf	G-05	CSE91
CSE9271	Ryan Uppal	G-53	CSE92
IT2256	Arnav Yogi	E-14	IT22
CSE9272	Amira Gera	G-12	CSE92
IT0554	Dhanuk Ramanathan	B-78	IT05
CSE0547	Kartik Joshi	B-22	CSE05

3 E-R Diagram



4 Programs

main.py

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

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

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

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

student.py

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

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

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

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

course.py

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

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

batch.py

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

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

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

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

examination.py

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

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

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

5 Outputs

Command Line Interface

```
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 1
    1. Create a new student
    2. Update details of a student
    3. Remove a student
    4. Generate report of a student
    : 1
                Student ID: CSE0547
                Student Name: Kartik Joshi
                Class Roll No: B-22
                Batch ID: CSE05
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: CSE9533
                Student Name: Umang De
                Class Roll No: B-04
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: ECE9994
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: CSE1878
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: H-55
                        Student Name: Vanya Suri
                        Marks: 89
                        Class Roll Number: C-42
                        Student Name: Anahita Kota
                        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='H-06', name='Elakshi
                        ↪  Chaudhry', marks=41.0)
                        Student(roll='C-21', name='Nitya Kari',
                        ↪  marks=100.0)
                        Student(roll='G-49', name='Khushi
                        ↪  Chatterjee', marks=99.0)
                        Student(roll='D-49', name='Dhanuk Deol',
                        ↪  marks=88.0)
                        Student(roll='H-62', name='Nehmat Dhillon',
                        ↪  marks=46.0)
                        Student(roll='E-66', name='Amani Handa',
                        ↪  marks=70.0)
                        Student(roll='G-18', name='Taimur Kibe',
                        ↪  marks=78.0)
                        Student(roll='F-06', name='Advik Kaur',
                        ↪  marks=85.0)
                        Student(roll='G-48', name='Sumer Kalita',
                        ↪  marks=83.0)
                        Student(roll='A-65', name='Hrishita Kunda',
                        ↪  marks=89.0)
                        Student(roll='H-89', name='Vardaniya
                        ↪  Subramanian', marks=78.0)
                        Student(roll='C-98', name='Taimur Dhaliwal',
                        ↪  marks=68.0)
                        Student(roll='C-14', name='Adah Contractor',
                        ↪  marks=84.0)
                        Student(roll='H-45', name='Yasmin Baral',
                        ↪  marks=69.0)
```

```
                        Student(roll='H-66', name='Sana Mann',
                         ↪  marks=73.0)
                        Student(roll='G-43', name='Dhanush Jhaveri',
                         ↪  marks=33.0)
                        Student(roll='A-56', name='Ahana  Agarwal',
                         ↪  marks=98.0)
                        Student(roll='D-42', name='Raghav Sachar',
                         ↪  marks=73.0)
                        Student(roll='G-01', name='Dharmajan Johal',
                         ↪  marks=89.0)
                        Student(roll='B-78', name='Dhanuk
                         ↪  Ramanathan', marks=87.0)
                        Student(roll='H-54', name='Zara Kunda',
                         ↪  marks=49.0)
                        Student(roll='B-71', name='Yashvi Sandal',
                         ↪  marks=98.0)
                        Student(roll='H-71', name='Amira Date',
                         ↪  marks=96.0)
                        Student(roll='F-33', name='Jiya Ratti',
                         ↪  marks=71.0)
                        Student(roll='H-56', name='Kavya Das',
                         ↪  marks=61.0)
                        Student(roll='H-80', name='Lagan Ahluwalia',
                         ↪  marks=53.0)
                        Student(roll='B-55', name='Zeeshan Bali',
                         ↪  marks=79.0)
                        Student(roll='E-14', name='Arnav Yogi',
                         ↪  marks=60.0)
                        Student(roll='G-07', name='Saanvi Divan',
                         ↪  marks=31.0)
                        Student(roll='D-18', name='Veer Chaudhari',
                         ↪  marks=47.0)
                        Student(roll='A-42', name='Uthkarsh Sen',
                         ↪  marks=90.0)
                        Student(roll='E-68', name='Shalv Kaur',
                         ↪  marks=90.0)
                        Student(roll='C-66', name='Chirag
                         ↪  Ramanathan', marks=46.0)
                        Student(roll='D-53', name='Pihu Agrawal',
                         ↪  marks=54.0)
                        Student(roll='E-82', name='Kiaan Sankaran',
                         ↪  marks=57.0)
                        Student(roll='E-94', name='Akarsh Vyas',
                         ↪  marks=67.0)
                        Student(roll='E-40', name='Adira Sengupta',
                         ↪  marks=76.0)
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: IT20
                Batch Name: IT 2000-2024
                Department Name: IT
                        Enter the courses for IT20
                                : C001
                                : C002
                                : C003
                                : C004
                                : C006
                                : C007
                                : C008
                                : C010
                                :
                        Enter the students for IT20
                                : IT2056
                                : IT0234
                                :
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: IT92
                 ['IT9262', 'IT9299', 'IT9258']
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: ECE10
                         ['C001', 'C002', 'C003', 'C004', 'C005',
                          ↪  'C006', 'C007', 'C008', 'C009', 'C010']
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 3
    1. Create a new batch
    2. View list of students in a batch
    3. View list of courses taught in a batch
    4. View performance of a batch
```

```
    5. Create pie chart of percentage of all students
    : 4
                Batch ID: CSE19
                        Student(roll='A-94', name='Damini Bhakta',
                        ↪  percentage=65.5)
                        Student(roll='D-44', name='Emir Rajagopal',
                        ↪  percentage=83.5)
                        Student(roll='A-18', name='Kiara Sundaram',
                        ↪  percentage=83.5)
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: CSE91
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 4
    1. Create a new department
    2. View batches of a department
    3. View average performance of batches of a department
    4. Create statistics of a department
    :1
                Department ID: BA
                Department Name: Business Administration
                        Enter the batches for BA
                                : BA22
                                : BA19
                                : BA19
                                :
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
                        ['CSE05', 'CSE07', 'CSE11', 'CSE12',
                            'CSE13', 'CSE15', 'CSE16', 'CSE18',
                            'CSE19', 'CSE21', 'CSE89', 'CSE90',
                        ↪↪↪  'CSE91', 'CSE92', 'CSE94', 'CSE95',
                        ↪↪↪  'CSE96', 'CSE98']
1. Student
2. Course
3. Batch
4. Department
5. Examination
: 4
    1. Create a new department
    2. View batches of a department
```

```
    3. View average performance of batches of a department
    4. Create statistics of a department
    :3
                Department ID: ECE
Performance(batch='ECE00', average=76.0)
Performance(batch='ECE02', average=76.73333333333333)
Performance(batch='ECE03', average=69.9)
Performance(batch='ECE04', average=62.6)
Performance(batch='ECE06', average=74.6)
Performance(batch='ECE08', average=72.0)
Performance(batch='ECE09', average=75.6)
Performance(batch='ECE10', average=51.8)
Performance(batch='ECE12', average=65.1)
Performance(batch='ECE13', average=62.5)
Performance(batch='ECE14', average=65.4)
Performance(batch='ECE22', average=70.6)
Performance(batch='ECE90', average=74.2)
Performance(batch='ECE91', average=74.2)
Performance(batch='ECE92', average=65.5)
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
                                : CSE21
                                : CSE89
                                :
Name of examination : Mid Semester
CSE21
        F-98
                C004: 89
                C007: 78
CSE89
        H-80
                C004: 63
                C007: 78
        C-19
                C004: 94
                C007: 56
    1. View student perfomance in the examination
    2. Create examination statistics
    : 1
[Performance(student_id='CSE2191', average=83.5),
    Performance(student_id='CSE8906', average=70.5),
 ⇨  Performance(student_id='CSE8941', average=75.0)]
1. Student
2. Course
3. Batch
4. Department
5. Examination
```

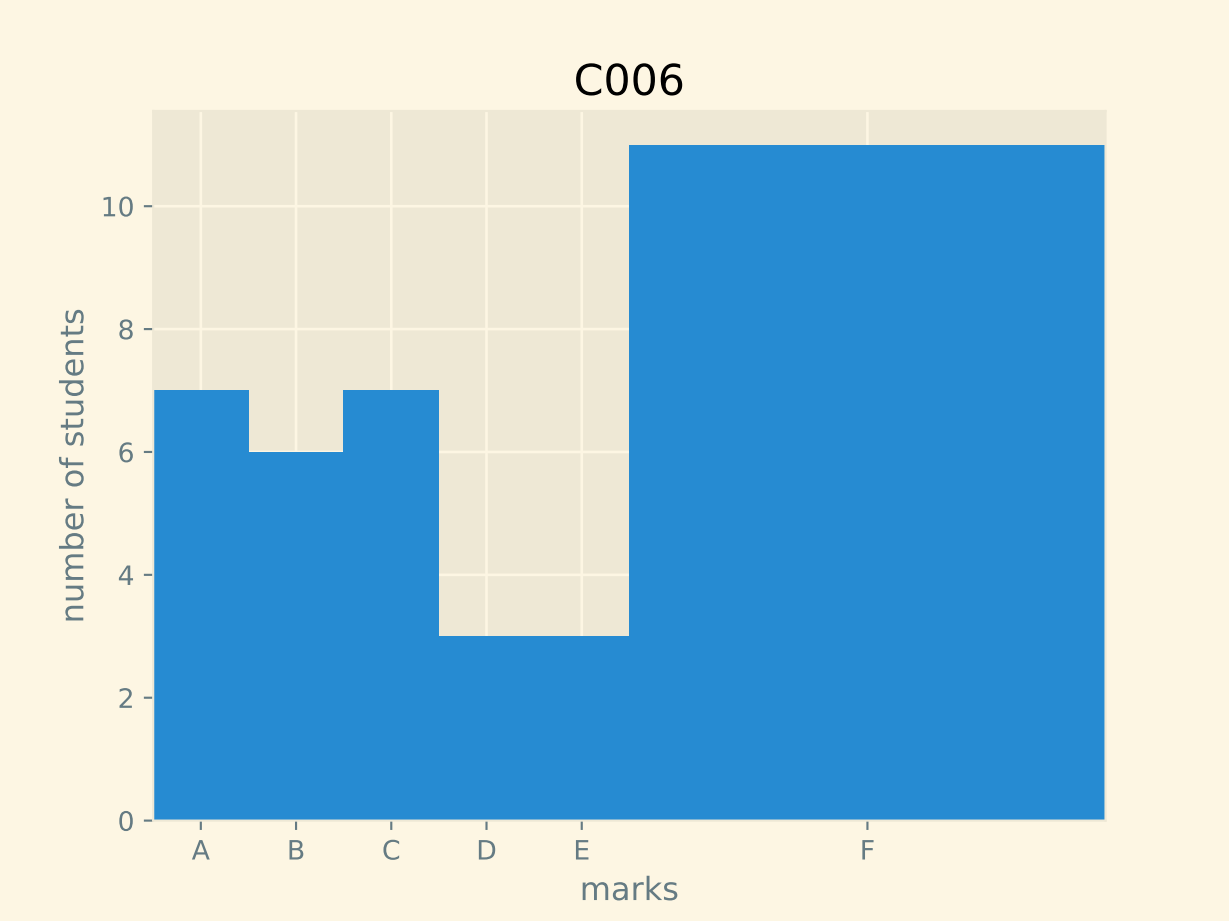
```
: 5
    Hold an examination:
                        Enter the batches for exam
                                : CSE05
                                : ECE02
                                : IT92
                                : CSE91
                                : IT22
                                :
Name of examination : End Semester
CSE05
        B-48
                C004: 94
                C007: 78
        D-20
                C004: 63
                C007: 48
        B-22
                C004: 59
                C007: 88
CSE91
        A-28
                C004: 44
                C007: 59
        H-89
                C004: 78
                C007: 63
        B-59
                C004: 47
                C007: 89
        G-05
                C004: 57
                C007: 59
ECE02
        G-49
                C001: 89
                C002: 76
                C003: 48
                C004: 98
                C005: 78
                C006: 98
                C007: 96
                C008: 74
                C009: 78
                C010: 96
        C-21
                C001: 98
                C002: 78
                C003: 69
                C004: 8
                C005: 78
                C006: 96
                C007: 37
                C008: 94
                C009: 76
                C010: 84
        D-49
                C001: 96
                C002: 79
                C003: 49
                C004: 75
                C005: 86
                C006: 96
                C007: 87
                C008: 94
                C009: 78
                C010: 86
IT22
        G-07
                C001: 97
                C002: 78
                C003: 96
                C004: 75
                C006: 25
                C007: 35
                C008: 48
                C010: 67
        E-14
```

```
                C001: 98
                C002: 48
                C003: 76
                C004: 48
                C006: 49
                C007: 67
                C008: 98
                C010: 78
IT92
        E-68
                C001: 98
                C002: 78
                C003: 94
                C004: 56
                C006: 78
                C007: 85
                C008: 86
                C010: 84
        C-66
                C001: 87
                C002: 89
                C003: 85
                C004: 82
                C006: 93
                C007: 94
                C008: 92
                C010: 93
        A-42
                C001: 89
                C002: 76
                C003: 84
                C004: 56
                C006: 37
                C007: 89
                C008: 84
                C010: 56
    1. View student perfomance in the examination
    2. Create examination statistics
    : 2
1. Student
2. Course
3. Batch
4. Department
5. Examination
:
```

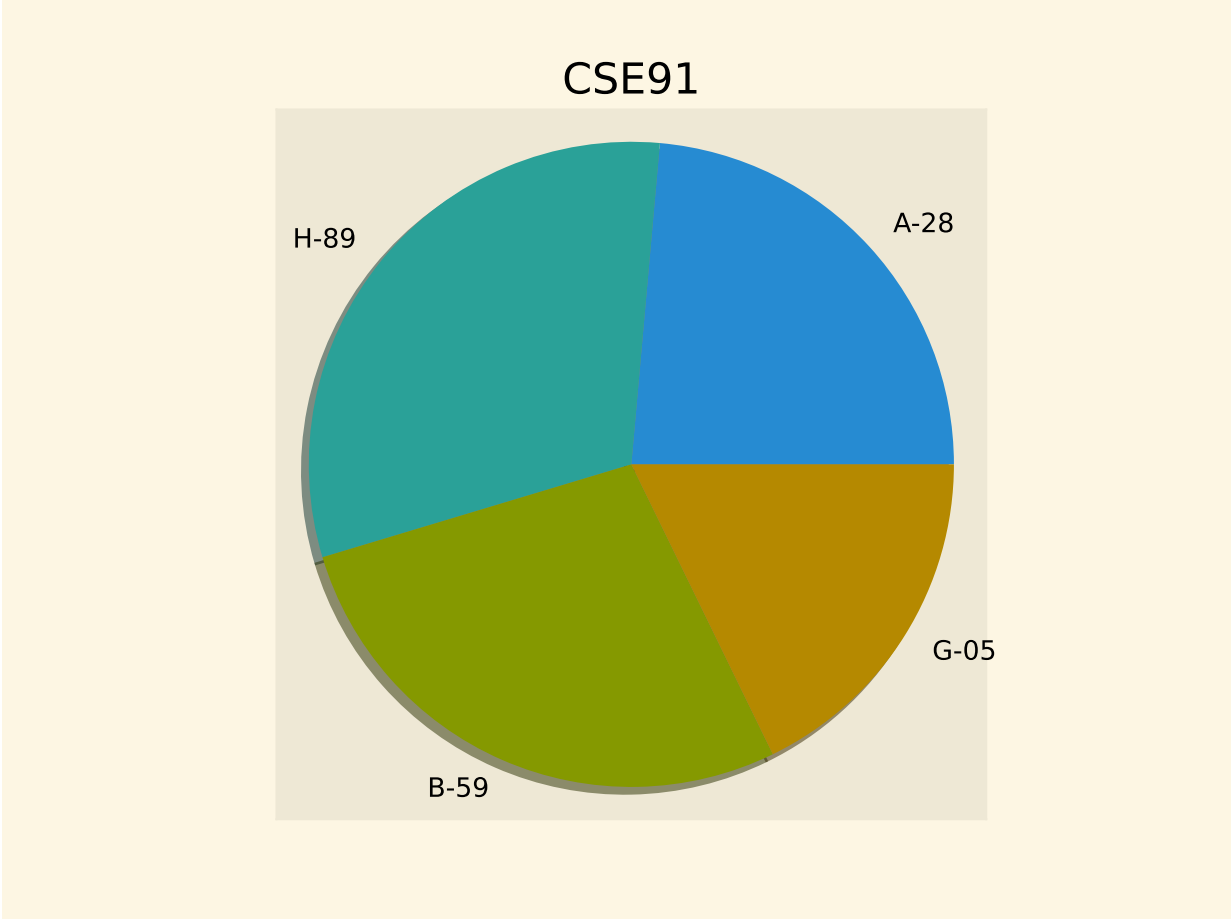
CSE1878-report_card.txt

```
Advika Kapur (C-59)

+------------+-----------+----------------+------------+-------+---------+
| Course     | Course Id | Marks Obtained | Full Marks | Grade | Remarks |
+------------+-----------+----------------+------------+-------+---------+
| Electrical | C004      |             67 |        100 |   D   | Passed  |
+------------+-----------+----------------+------------+-------+---------+
| Design     | C007      |             89 |        100 |   B   | Passed  |
+------------+-----------+----------------+------------+-------+---------+
| Total      | -         |            156 |        200 |   C   | Passed  |
+------------+-----------+----------------+------------+-------+---------+

ID:CSE1878
```

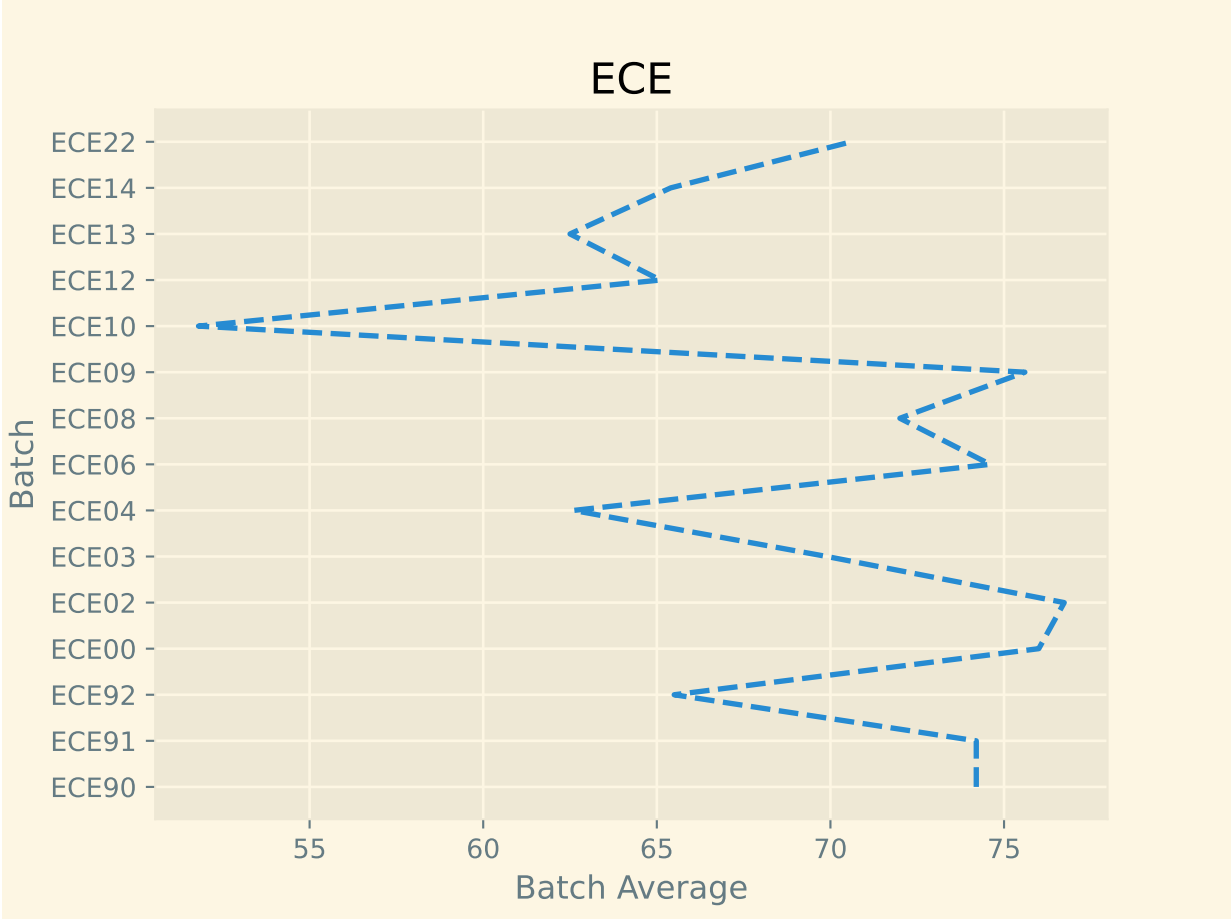
Course Statistics-C006.pdf

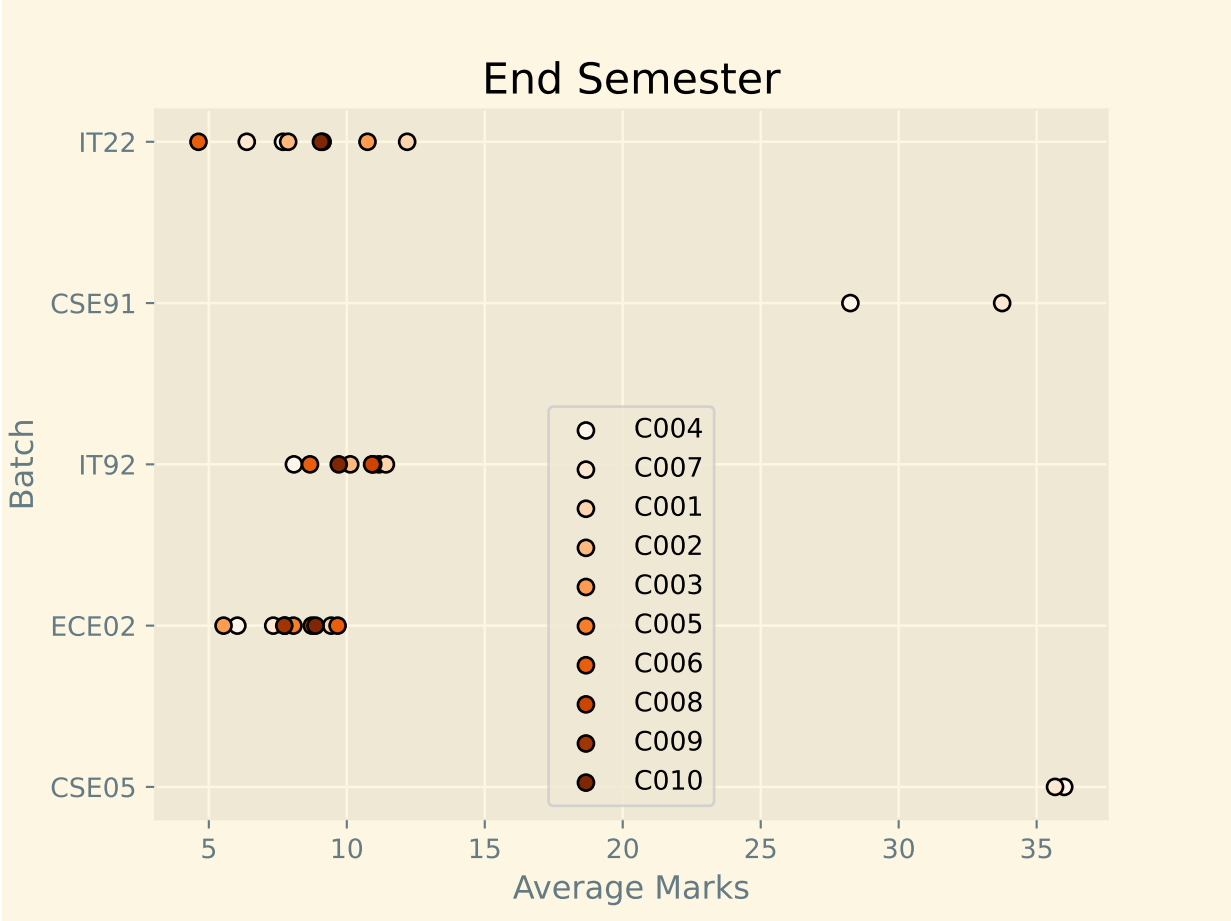


Batch Statistics-CSE91.pdf



Department Statistics-ECE.pdf




End Semester
IT22
CSE91
IT92
ECE02
CSE05
Batch
C004
C007
C001
C002
C003
C005
C006
C008
C009
C010
5
10
15
20
25
30
35
Average Marks