**BIG DATA ANALYTICS (3170722)**



**VVP**

**Engineering**

**College**

Submitted by: **DISHEN MAKWANA**

**180470107035**

**G2**

 **V. V. P. Engineering College, Rajkot**

**Department of Computer Engineering**

**Vision of the Institute**

* To be an exemplary institute, transforming students into competent

professionals with human values.

**Mission of the Institute**

* To provide a conducive academic environment for strengthening technical capabilities of the students.
* To strengthen linkage with industries, alumni and professional bodies.
* To organize various co-curricular and extra-curricular activities for overall development of the students.
* To practice good governance and conduct value- based activities for making students responsible citizens.

**Vision of the Department**

* Transforming students into globally efficient professionals with moral values.

**Mission of the Department**

* To provide a strong foundation of computer engineering through effective teaching learning process.
* To enhance industry linkage & alumni network for better placement and real-world exposure.
* To provide various opportunities & platforms for all round development of students & encourage them for value-based practices.

**Program Educational Objectives (PEOs)**

Graduates will be able to

* Apply computer engineering theories, principles and skills to meet the challenges of the society.
* Communicate effectively, work collaboratively and manifest professionalism with ethics.
* Exhibit life-long learning attitude and adapt to rapid technological changes in industry.
* Advance their career in industry, pursue higher education or become an entrepreneur.

A picture containing icon

Description automatically generated

## V.V.P. ENGINEERING COLLEGE

**RAJKOT**

**Certificate**

This is to certify that

Mr. DISHEN MAKWANA, Enrollment No: 180470107035, Branch: Computer Engineering, Semester: 7 has satisfactorily completed the course in the subject: **BIG DATA ANALYTICS (3170722)** within the four walls of V.V.P. Engineering College, Rajkot.

Date of Submission:

**Dr. Maulik Dhamecha**, Head of Department,

Staff In-Charge Department of Computer Engineering,

V.V.P. Engineering College

A picture containing text, clipart

Description automatically generated**V. V. P. Engineering College**

**Department of Computer Engineering**

**Course Outcomes**

Semester: 7th

Subject: BIG DATA ANALYTICS

Code:3170722

After learning the course, the students will be able to

|  |  |
| --- | --- |
| **Sr. No.** | **CO Statement** |
| 1 | identify big data application areas |
| 2 | use big data framework |
| 3 | model and analyze data by applying selected techniques |
| 4 | demonstrate an integrated approach to big data |

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Tutorial** | **Page**  **No.** | **Sign** |
| 1 | **Practical 1:** Study about different Applications of Big data Analytics. | **6** |  |
| 2 | **Practical 2:** Study about different Applications of Structured, semi structured, and un-structured database. | **7** |  |
| 3 | **Practical 3:** Write a program to get average marks by entering student name or enrollment number. | **9** |  |
| 4 | **Practical 4:** Write a program for word count using Map Reduce algorithm (Using Static database). | **12** |  |
| 5 | **Practical 5:** Write a program for word count using Map Reduce algorithm (Using dynamic database). | **14** |  |
| 6 | **Practical 6:** Write a program for word count using Map Reduce algorithm and generate every stage output using key-value pair. | **16** |  |
| 7 | **Practical 7:** Write a program to add column in existing table and get average marks by entering student name or enrollment number. | **18** |  |
| 8 | **Practical 8:** Study about basic CRUD operations in MongoDB. | **22** |  |

**Practical 1**

1. **LIST 5 APPLICATIONS OF BIG DATA ANALYSIS.**
2. **YOUTUBE:**

* More than 400 hours of video content is uploaded on YouTube every minute, and approximately 1 billion hours of YouTube videos are watched every day.
* This amount of huge data is way beyond the scope of normal database tools to analyses it.
* Thus with billions of users YouTube surely uses big data Analysis to recommend content based on our history, liked video ,View counts and
* more such data.

1. **Image Processing to generate image of event horizon:**

* The first ever image of the event horizon of M87 black hole was produced. With 7 telescopes that produced less almost equivalent to size of earth took.
* The resulting quadrillions of bytes of data took two years to process before they could form the image.
* This another use of Big data Analysis that produced incredible results.

1. **COVID-19 TRACKING APP:**

* Arogya Setu app also requires data such as full name, gender, age, travel history of the last 30 days, and professional details.
* Aarogya Setu also entails a self-assessment questionnaire such as cough, fever or difficult breathing symptoms, diabetes, hypertension, lung/ heart disease.
* International travel over the last 14 days. Questions pertaining to recent interactions with COVID-positive individuals if you are a healthcare professional and have examined a COVID-positive case without protective gear.
* This surely processes huge data to help track red zones and help to reduce risk of spreading covid.

1. **GOVERNMENT WEBSITES**:

* These websites have information of aadhar card, pan cards, licences , income tax returns etc of crores of Indian citizens , these produce huge amounts of data to handle.
* Thus, we can say that they produce big data.

1. **Large Hadron Collider**:

* The Large Hadron Collider experiments represent about 150 million sensors delivering data 40 million times per second.
* There are nearly 600 million collisions per second. If all sensor data were recorded in LHC, the data flow would be extremely hard to work with.
* The data flow would exceed 150 million petabytes annual rate, or nearly 500 exabytes per day, before replication. To put the number in perspective, this is equivalent to 500 quintillion (5×1020) bytes per day, almost 200 times more than all the other sources combined in the world.

**Practical 2**

1. **LIST EXAMPLES OF TYPES OF DATABASES**

**Unstructured Database Examples**

1. **Internet of Things (IoT). Sensor data**
2. **Nasa’s Space research data**

* NASA has a strong track record of archiving and providing universal access to science data products from its science missions and programs.
* As a matter of longstanding policy and practice, NASA archives all science mission data products to ensure long-term usability and to promote wide-spread usage by scientists, educators, decision-makers, and the public.
* Our vision is to facilitate the on-going scientific discovery process and inspire the public through the body of knowledge captured in these public archives.
* The archives are primarily organized by science discipline or theme. Communities of practice within these disciplines and themes are actively engaged in the planning and development of archival capabilities to ensure responsiveness and timely delivery of data to the public from the science missions.

1. **Scientific data**

* Here can be given many unstructured data examples: oil gas exploration, seismic imagery, atmospheric data, space exploration and so on.

1. **Social media**

* Social media has become huge data generator, it is the preferred channel when it comes to viewing, creating, or sharing information.
* Social media is also used by businesses, governments, and organizations across domains such as shopping, entertainment, education, crisis management, and politics.
* Social media platforms generate data at every moment, round-the-clock, all over the world. This has led to a huge data that could be in the form of text, images, videos, audio, or geo-locations.

1. **Survey responses**

* A questionnaire to conduct market research or employee engagement typically includes multiple-choice and open-ended questions.
* Responses to such open-ended questions are unstructured text. These questions are important as they help the researcher discover aspects they may not have considered, gain a deeper understanding, and build a connection with the respondent.

**Sources of semi-structured Data**

1. **Email:**

* Email messages are a good example. While the actual content is unstructured, it does contain structured data such as name and email address of sender and recipient, time sent, etc.

1. **Digital Photography:**

* The image itself is unstructured, but if the photo was taken on a smart phone, for example, it would be date and time stamped, geo tagged, and would have a device ID.

1. **NoSQL Database:**

* **NoSQL databases store data in documents rather than relational tables.**
* Accordingly, we classify them as "not only SQL" and subdivide them by a variety of flexible data models.
* Types of NoSQL databases include pure document databases, key-value stores, wide-column databases, and graph databases. NoSQL databases are built from the ground up to store and process vast amounts of data at scale and support a growing number of modern businesses.

1. **CSV, XML and JSON documents**
2. **Electronic data interchange (EDI)**

**Structured Database**

1. **Login System**
2. **Library System**
3. **Billing System**
4. **Banking System**
5. **GTU Student Portal System**

**Practical 3**

**Display the Average Marks of Student by retrieving data from the database based on input from the user.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | ENROLL | MARK1 | MARK2 | AVG |
| John | CE044 | 16 | 18 | 17 |
| Doe | CE045 | 12 | 10 | 11 |
| Harsh | CE046 | 14 | 14 | 14 |
| Jugal | CE047 | 10 | 16 | 13 |
| Dishen | CE048 | 8 | 12 | 10 |
| Rutvik | CE049 | 10 | 8 | 9 |
| Parth | CE050 | 14 | 10 | 12 |
| Apurv | CE051 | 12 | 14 | 13 |
| Darshan | CE052 | 16 | 18 | 17 |
| Dhruv | CE053 | 18 | 12 | 15 |

**Program :**

#include "bits/stdc++.h"

using namespace std;

class Student

{

public:

string name;

string er\_no;

int mark1;

int mark2;

Student(string name, string er\_no, int mark1, int mark2)

{

this->name = name;

this->er\_no = er\_no;

this->mark1 = mark1;

this->mark2 = mark2;

}

};

double search(string key, vector<Student> &students)

{

for (auto x : students)

{

if (x.name == key)

{

return (double)(x.mark1 + x.mark2) / 2.0;

}

if (x.er\_no == key)

{

return (double)(x.mark1 + x.mark2) / 2.0;

}

}

return -1;

}

int32\_t main()

{

vector<Student> students;

students.push\_back(Student("John", "CE044", 98, 95));

students.push\_back(Student("Doe", "CE045", 95, 95));

students.push\_back(Student("Harsh", "CE046", 90, 95));

students.push\_back(Student("Jugal", "CE047", 98, 90));

students.push\_back(Student("Dishen", "CE048", 99, 98));

students.push\_back(Student("Rutvik", "CE049", 90, 90));

students.push\_back(Student("Parth", "CE050", 90, 92));

students.push\_back(Student("Apurv", "CE051", 91, 93));

students.push\_back(Student("Krunal", "CE052", 100, 95));

students.push\_back(Student("Jay", "CE053", 96, 99));

string key;

cout << "Enter key value for search : " << endl;

cin >> key;

double ans = search(key, students);

if (ans == -1)

{

cout << "Key not found" << endl;

}

else

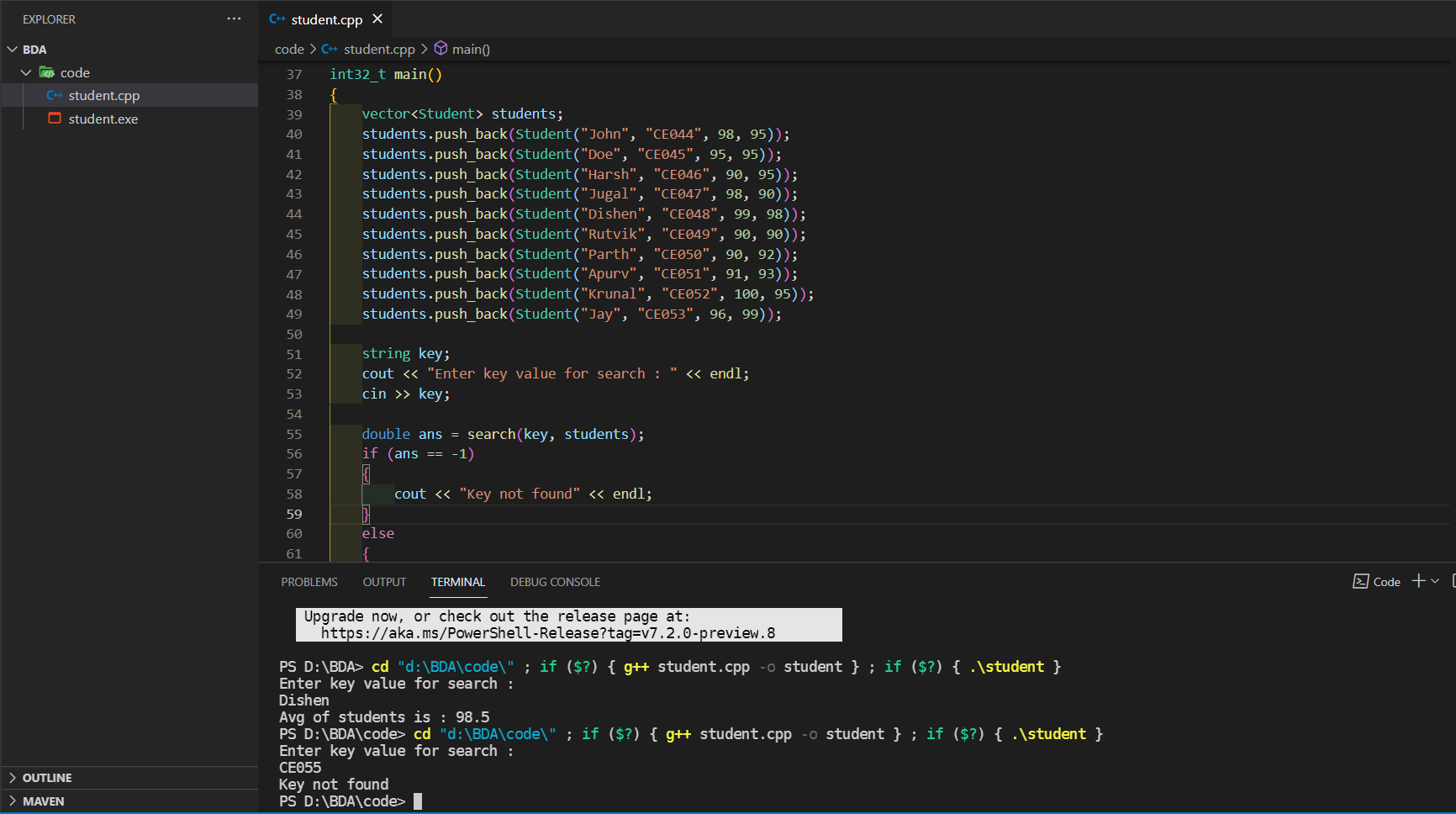
{

cout << "Avg of students is : " << ans << endl;

}

return 0;

}



**Practical 4**

**Write a program for word count using Map Reduce algorithm (Using Static database).**

**Program:**

#include "bits/stdc++.h"

using namespace std;

map<string, int> mp;

void mapping(string s)

{

istringstream ss(s);

string word;

while (ss >> word)

{

mp[word]++;

}

}

void reducer()

{

for (auto x : mp)

{

cout << x.first << " : " << x.second << endl;

}

}

int32\_t main(int argc, char \*argv[])

{

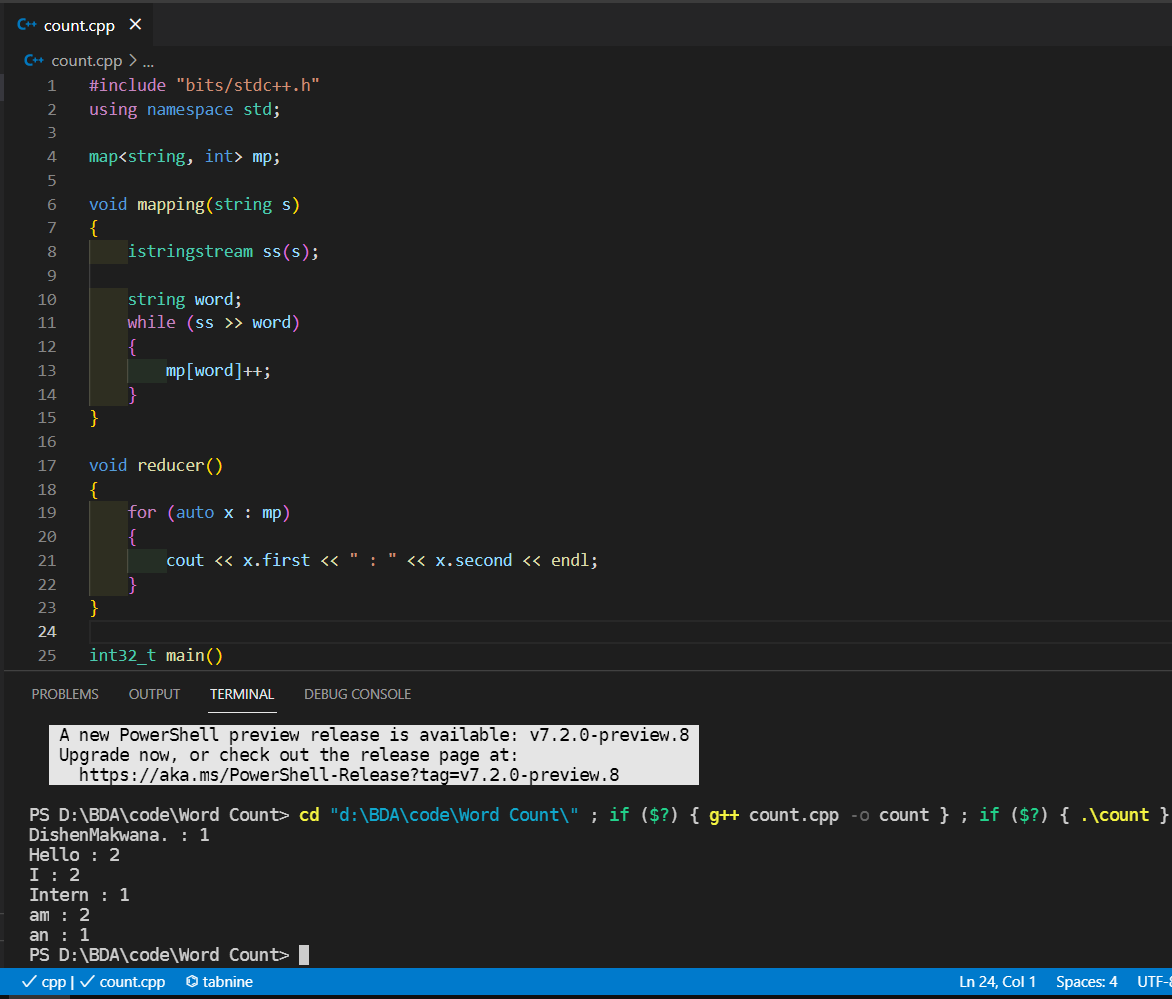
string s = "Hello I am11 DishenMakwana. Hello I am an Intern";

mapping(s);

reducer();

return 0;

}



**Practical 5**

**Write a program for word count using Map Reduce algorithm (Using dynamic database).**

**Program:**

#include "bits/stdc++.h"

using namespace std;

map<string, int> mp;

void mapping(string s)

{

istringstream ss(s);

string word;

while (ss >> word)

{

mp[word]++;

}

}

void reducer()

{

for (auto x : mp)

{

cout << x.first << " : " << x.second << endl;

}

}

int32\_t main(int argc, char \*argv[])

{

string line, file;

cin >> file;

ifstream myfile(file);

// string s = "Hello I am DishenMakwana. Hello I am an Intern";

if (myfile.is\_open())

{

while (getline(myfile, line))

{

mapping(line);

}

myfile.close();

}

else

{

cout << "Unable to open file" << endl;

return 0;

}

reducer();

return 0;

}

**Input file:**

Hello I am DishenMakwana. Hello I am an Intern

**Output:**

DishenMakwana : 1

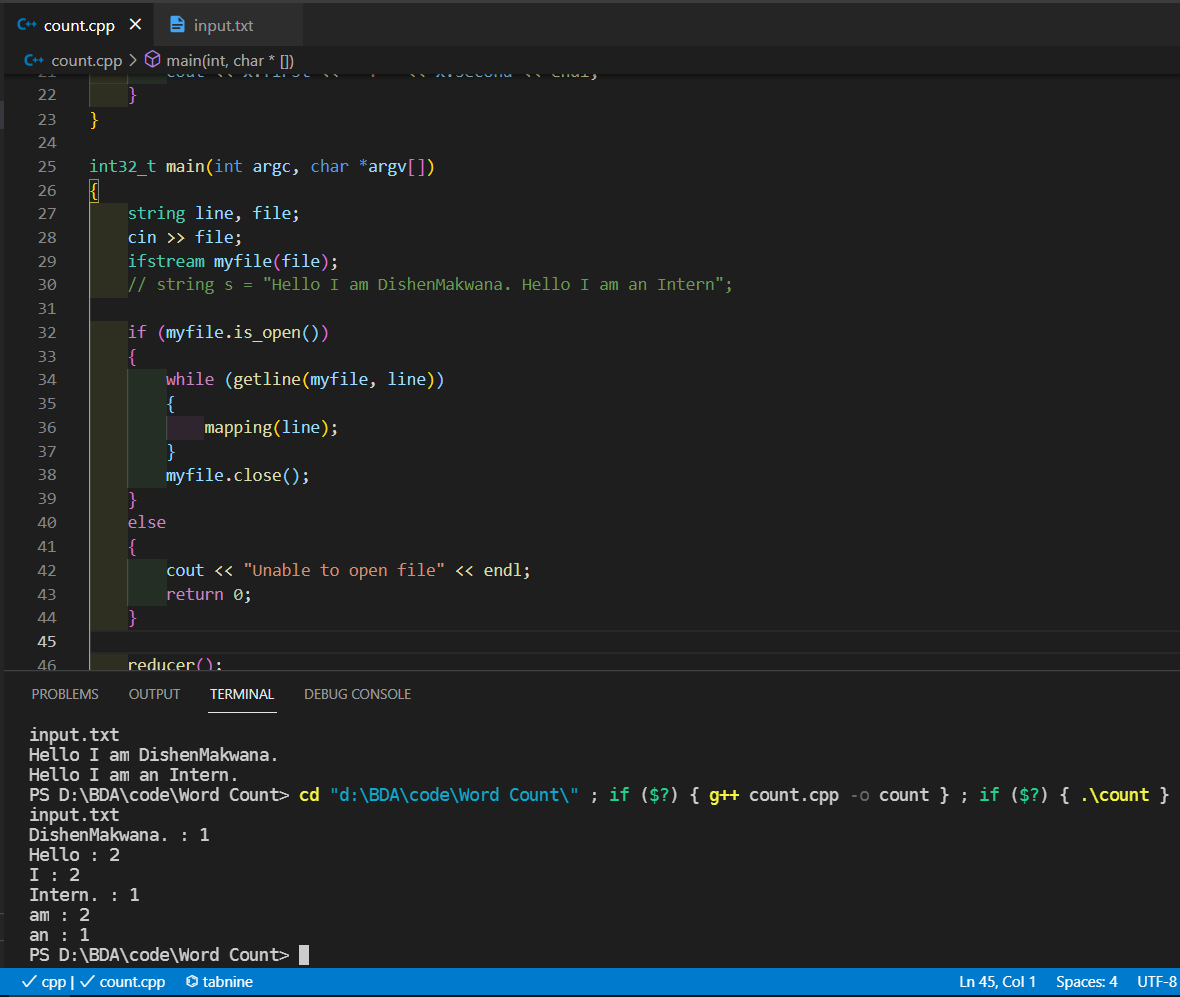
Hello : 2

I : 2

Intern : 1

am : 2

an : 1



**Practical 6**

**Write a program for word count using Map Reduce algorithm and generate every stage output using key-value pair.**

**Program:**

#include "bits/stdc++.h"

using namespace std;

unordered\_map<string, int> mp;

void mapping(string s)

{

istringstream ss(s);

string word;

while (ss >> word)

{

mp[word]++;

}

}

void reducer()

{

for (auto x : mp)

{

cout << x.first << " : " << x.second << endl;

}

}

int32\_t main(int argc, char \*argv[])

{

string line, file;

cin >> file;

ifstream myfile(file);

if (myfile.is\_open())

{

while (getline(myfile, line))

{

mapping(line);

}

myfile.close();

}

else

{

cout << "Unable to open file" << endl;

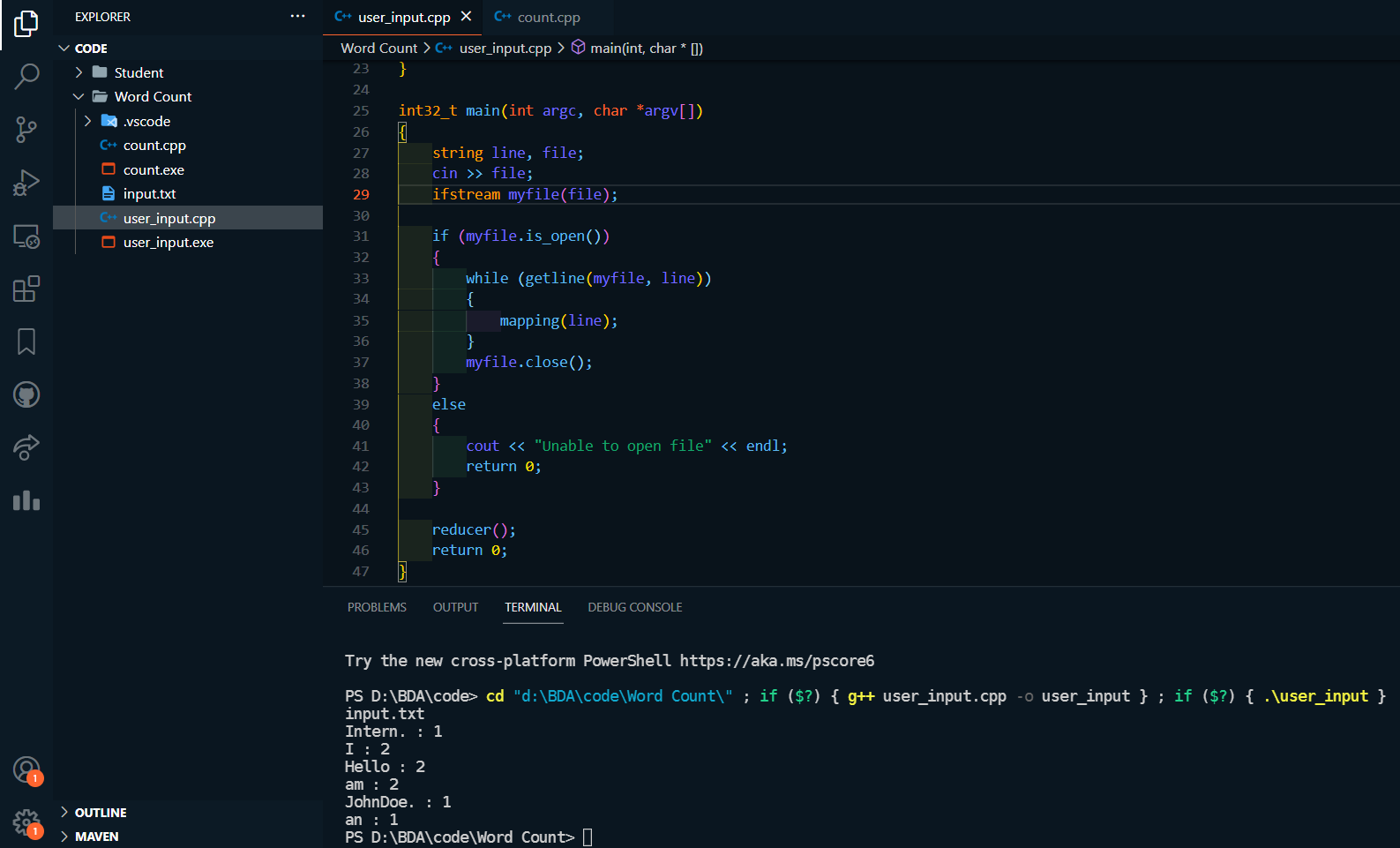
return 0;

}

reducer();

return 0;

}



Input.txt file:

Hello I am John Doe.

Hello, I am an Intern.

**Practical 7**

**Write a program to add column in existing table and get average marks by entering student name or enrollment number.**

#include "bits/stdc++.h"

using namespace std;

class Student

{

public:

string name;

string er\_no;

int mark1;

int mark2;

Student(string name, string er\_no, int mark1, int mark2)

{

this->name = name;

this->er\_no = er\_no;

this->mark1 = mark1;

this->mark2 = mark2;

}

};

class MasterStudent : public Student

{

public:

int mark3;

MasterStudent(string name, string er\_no, int mark1, int mark2, int mark3) : Student(name, er\_no, mark1, mark2)

{

this->mark3 = mark3;

}

};

double search(string key, vector<Student> &students)

{

for (auto x : students)

{

if (x.name == key)

{

return (double)(x.mark1 + x.mark2) / 2.0;

}

if (x.er\_no == key)

{

return (double)(x.mark1 + x.mark2) / 2.0;

}

}

return -1;

}

double search(string key, vector<MasterStudent> &ms)

{

for (auto x : ms)

{

if (x.name == key)

{

return (double)(x.mark1 + x.mark2 + x.mark3) / 3.0;

}

if (x.er\_no == key)

{

return (double)(x.mark1 + x.mark2 + x.mark3) / 3.0;

}

}

return -1;

}

int32\_t main()

{

vector<Student> students;

students.push\_back(Student("John", "CE044", 98, 95));

students.push\_back(Student("Mary", "CE045", 95, 91));

students.push\_back(Student("Doe", "CE046", 90, 93));

students.push\_back(Student("Harsh", "CE047", 94, 97));

students.push\_back(Student("Raj", "CE048", 91, 92));

students.push\_back(Student("Jugal", "CE049", 92, 93));

students.push\_back(Student("Rutvik", "CE050", 93, 94));

students.push\_back(Student("Parth", "CE051", 94, 95));

students.push\_back(Student("Krunal", "CE052", 95, 96));

students.push\_back(Student("Dishen", "CE053", 98, 97));

string s;

cout << "You want to add extra column?" << endl;

cin >> s;

vector<MasterStudent> master\_students;

if (s == "Yes" || s == "yes")

{

string key;

int data;

cout << "Enter the name of the student : " << endl;

cin >> key;

cout << "Enter the mark of the student : " << endl;

cin >> data;

for (auto x : students)

{

if (x.name == key)

{

master\_students.push\_back(MasterStudent(x.name, x.er\_no, x.mark1, x.mark2, data));

}

else

{

master\_students.push\_back(MasterStudent(x.name, x.er\_no, x.mark1, x.mark2, 0));

}

}

}

string key;

cout << "Enter key value for search : " << endl;

cin >> key;

double ans;

if (s == "Yes" || s == "yes")

{

ans = search(key, master\_students);

}

else

{

ans = search(key, students);

}

if (ans == -1)

{

cout << "Key not found" << endl;

}

else

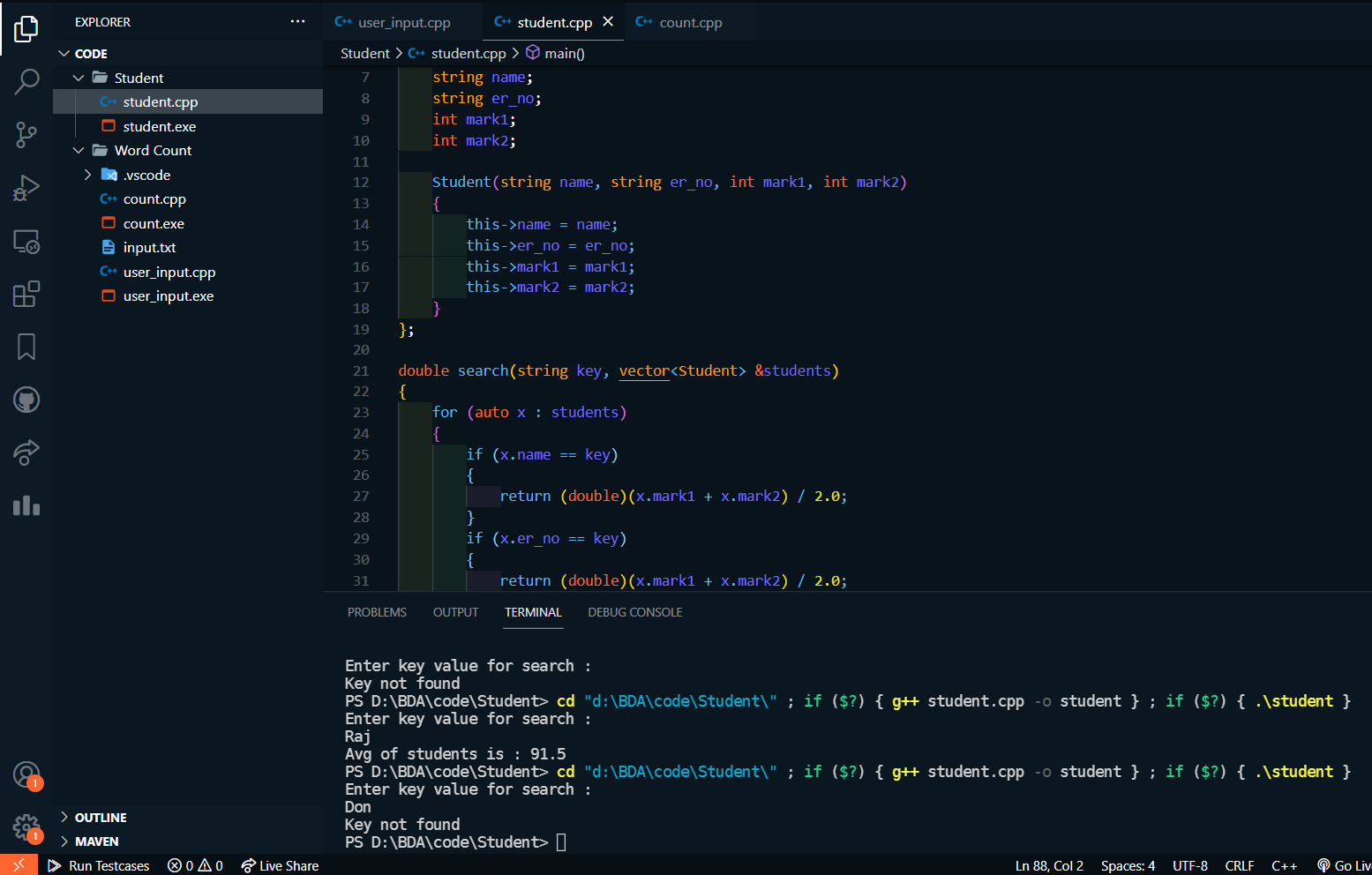
{

cout << "Avg of students is : " << ans << endl;

}

return 0;

}



**Practical 8**

**Study about basic CRUD operations in MongoDB.**

**C -> Create**

**R -> Read**

**U -> Update**

**D -> Delete**

**Create Operations**

The create or insert operations are used to insert or add new documents in the collection. If a collection does not exist, then it will create a new collection in the database. You can perform, create operations using the following methods provided by the MongoDB:

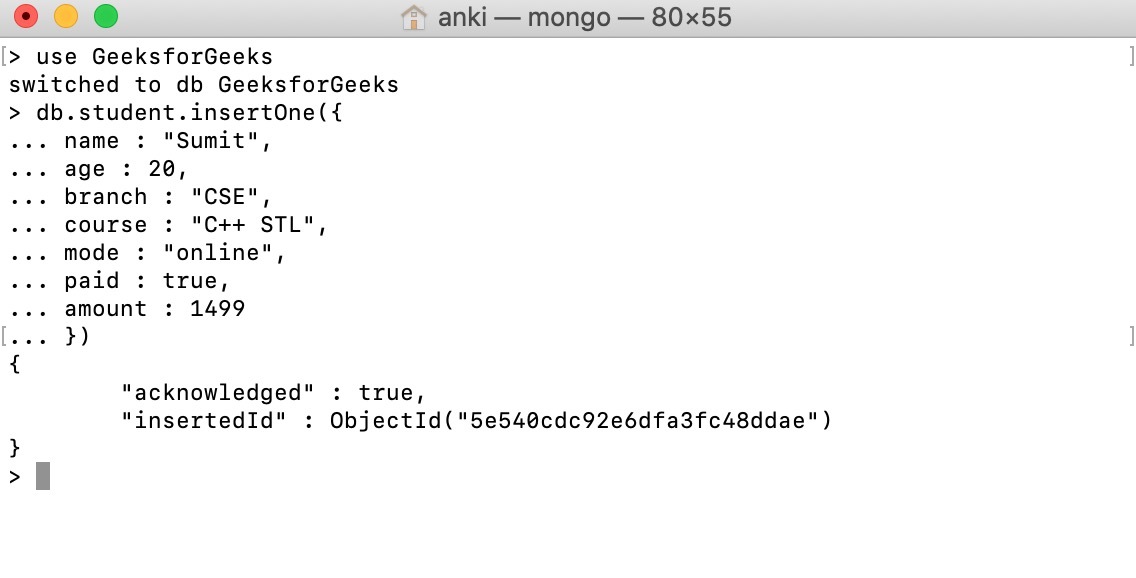
**db.collection.insertOne()**

It is used to insert a single document in the collection.

**db.collection.insertMany()**

It is used to insert multiple documents in the collection.

**Example**

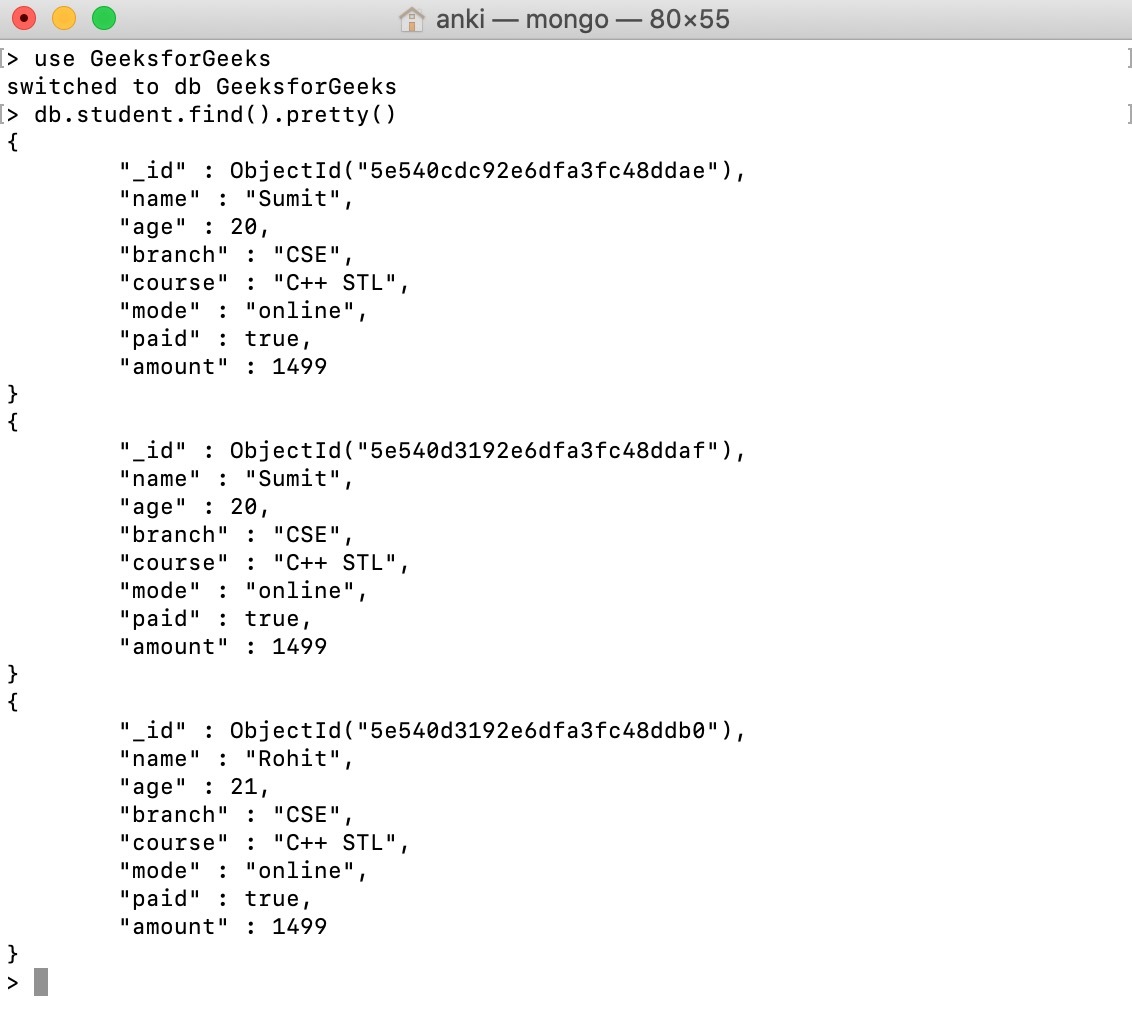


**Read Operations**

The Read operations are used to retrieve documents from the collection, or in other words, read operations are used to query a collection for a document. You can perform read operation using the following method provided by the MongoDB:

**db.collection.find()**

It is used to retrieve documents from the collection.



**Update Operations**

The update operations are used to update or modify the existing document in the collection. You can perform update operations using the following methods provided by the MongoDB:

**db.collection.updateOne()**

It is used to update a single document in the collection that satisfy the given criteria.

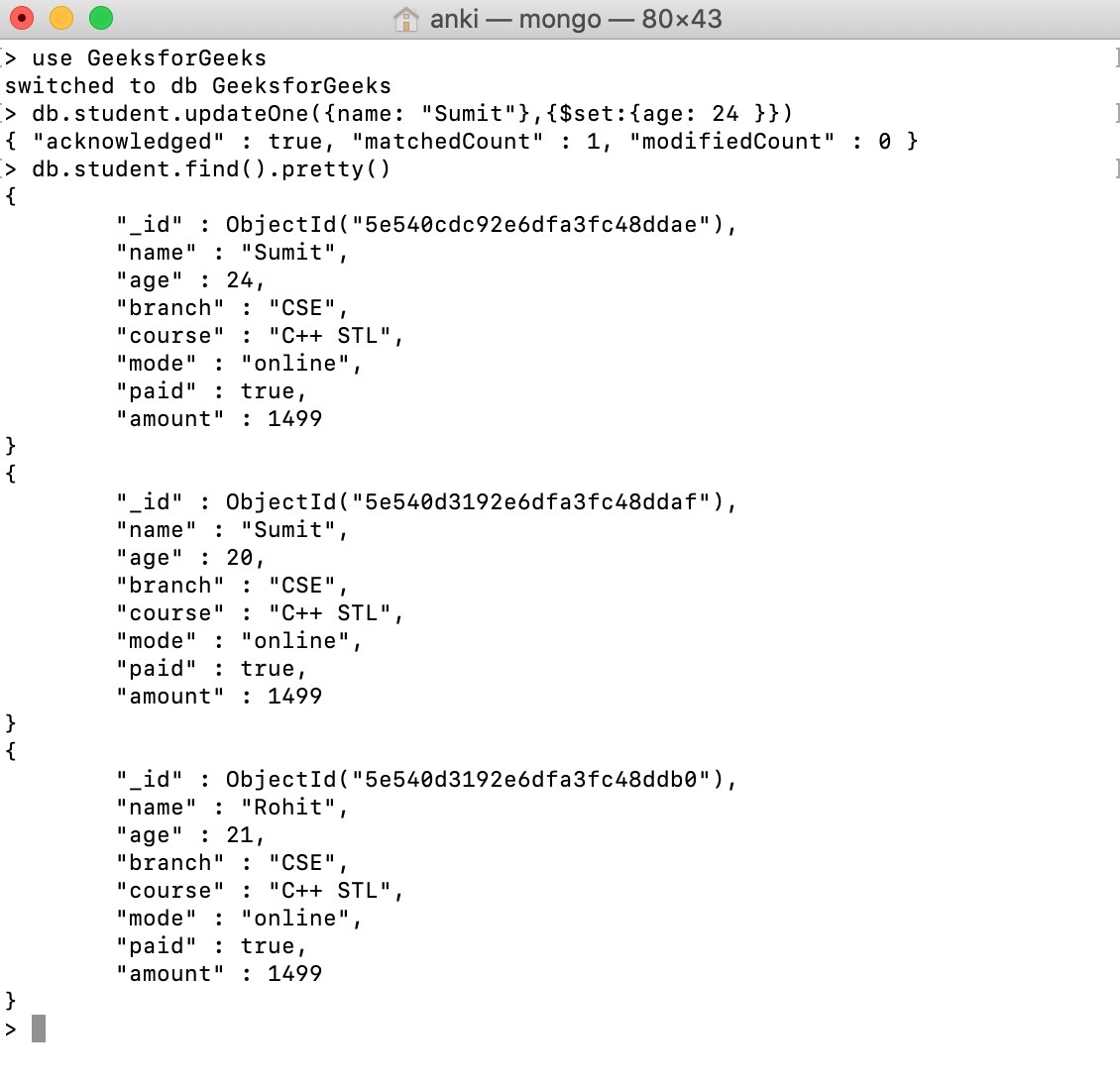
**db.collection.updateMany()**

It is used to update multiple documents in the collection that satisfy the given criteria.

**db.collection.replaceOne()**

It is used to replace single document in the collection that satisfy the given criteria.

**Example**



**Delete Operations**

The delete operation are used to delete or remove the documents from a collection. You can perform delete operations using the following methods provided by the MongoDB:

**db.collection.deleteOne()**

It is used to delete a single document from the collection that satisfy the given criteria.

**db.collection.deleteMany()**

It is used to delete multiple documents from the collection that satisfy the given criteria.

**Example**

A picture containing table

Description automatically generated