

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
BELAGAVI, KARNATAKA, INDIA**



**A MINI-PROJECT REPORT**

**ON**

**“UNIVERSITY MANAGEMENT SYSTEM”**

**Submitted in partial fulfillment of the requirement for the VI semester BE in**

**Information Science and Engineering**

**FS mini-project-18ISL67**

**Submitted by**

**ROHITH GANAPATHI BHAT [1SG19IS086]  
S.KAUSHAL [1SG19IS087]**

**Under the guidance of**

**Prof. Ambika S  
Assistant Professor**



**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

**SAPTHAGIRI COLLEGE OF ENGINEERING**

**(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)**

**#14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru – 560057**

**(Accredited by NBA and NAAC with 'A' GRADE)**

**(An ISO 9001:2015 & ISO 14001:2015 Certified Institution)**

**2021-22**

# SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)

#14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru – 560057

(Accredited by NBA and NAAC with 'A' GRADE)

(An ISO 9001:2015 & ISO 14001:2015 Certified Institution)

## Department of Information Science & Engineering



## CERTIFICATE

Certified that the project work entitled “**UNIVERSITY MANAGEMENT SYSTEM**” carried out by **ROHITH[1SG19IS086]**, **KAUSHAL[1SG19IS087]** bonafide students of 6<sup>th</sup> semester, department of **Information Science & Engineering** carried out at our college **Sapthagiri College of Engineering**, Bengaluru in partial fulfillment for the 6<sup>th</sup> Semester BE, FILE STRUCTURES Mini-Project-18ISL67 by **Visvesvaraya Technological University**, Belagavi during the year 2021-22. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Mini-Project work prescribed for the said Degree.

---

**Signature of the Guide**  
**Prof. Ambika S**  
**Assistant Professor**

---

**Signature of the HOD**  
**Dr H R Ranganatha**  
**HOD, I.S.E., S.C.E.**

### External Viva

**Name of the Examiners**

**Signature of the Examiners with date**

1. ....  
2. ....

1.....  
2.....

## ACKNOWLEDGEMENT

Any achievement doesn't depend solely on the individual efforts but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities have helped us. We would like to take this opportunity to thank them all.

We would like to express our heart-felt gratitude to **Dr. H Ramkrishna**, Principal, Sapthagiri College of Engineering, Bengaluru, for his help and inspiration during the tenure of the course.

It is great privilege to extend our deep sense of gratitude to **Dr. H R Ranganatha**, Head of the Department, Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru, who patronized throughout our career, for his constant support and encouragement and for the facilities provided to carry out this work successfully.

We wish to express our sincere thanks to our guide **Prof. Ambika S**, Assistant Professor, Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for helping us throughout and guiding us from time to time.

We also extend our sense of gratitude and sincere thanks to all faculty members and non-teaching staff members of Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for their views and encouraging ideas.

Finally, we also thank our family and friends for their co-operation and motivation.

**ROHITH (1SG19IS086)**

**KAUSHAL (1SG19IS087)**

# LIST OF FIGURES

FIGURE NUMBER	NAME OF FIGURE	PAGE NUMBER
3.2.1	APPLICATION DASHBOARD	16
3.2.2	INSERTION OF RECORD	16
3.2.3	DELETION OF RECORD	17
3.2.4	DISPLAY OF RECORD	18
3.2.5	SEARCHING OF A RECORD	18
3.2.6	MODIFICATION OF A RECORD	18
3.2.7	INSERTION OF RECORD ERROR	19
3.2.8	DELETION OF A RECORD ERROR	19
3.2.9	MODIFICATION OF A RECORD ERROR	20
3.2.10	SEARCHING OF A RECORD ERROR	20
4.4.1	BEFORE AND AFTER INSERTION	24
4.4.2	BEFORE AND AFTER DELETION	25
4.4.3	BEFORE AND AFTER MODIFICATION	26
4.4.4	FILE CONTENT	27
4.4.5	INDEX FILE CONTENT	27

# TABLE OF CONTENTS

Chapter no	Chapter name	Page no
<b>1</b>	<b>INTRODUCTION</b>	1-5
1.1	Introduction to File Structure	1
1.1.1	History	2
1.1.2	About the File	3
1.1.3	Various Storage Kind of Fields and Records	3-4
1.1.4	Application of File Structure	4-5
<b>2</b>	<b>SYSTEM ANALYSIS</b>	6-8
2.1	Analysis of Application	6
2.2	Structure used to Store the Fields and Records	6
2.3	Operation Performed on a File	6-7
2.4	Indexing Used	8
2.5	Compression Used	8
<b>3</b>	<b>SYSTEM DESIGN</b>	9-14
3.1	DESIGN OF FIELDS AND RECORDS	9
3.2	User Interface	9
3.2.1	Application Dashboard	10
3.2.2	Insertion of a Record	10
3.2.3	Deletion of a Record	11
3.2.4	Display of a Record	11
3.2.5	Searching of a Record	12
3.2.6	Modification of a Record	12
3.2.7	Insertion of a Record Error	13
3.2.8	Deletion of a Record Error	13
3.2.9	Modification of a Record Error'	14

3.2.10	Searching of a Record Error	14
<b>4</b>	<b>IMPLEMENTATION</b>	15-21
4.1	About Python	15
4.2	Pseudo code	15
4.2.1	Insertion operation pseudo code	16
4.2.2	Display operation pseudo code	16
4.2.3	Update operation pseudo code	16
4.2.4	Deletion operation pseudo code	16
4.2.5	Search operation pseudo code	16
4.3	TESTING	17
4.3.1	Unit Testing	17
4.3.2	System Testing	18
4.3.3	Integration Testing	18
4.4	Discussion of Result	19
4.4.1	Before and After Insertion	19
4.4.2	Before and After Deletion	20
4.4.3	Before and After Modification	21
4.4.4	File contents	22
4.4.5	Index file contents	22
<b>5</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	23
<b>6</b>	<b>REFERENCE</b>	24

## ABSTRACT

It's very important to keep our self up to date for new technologies. Nowadays every field is digitalized and the storage of data become very easy. We were using papers and registers to store the data in the many fields like educational fields,medical fields etc.But as many storage structures are came into existence it become very easy to maintain the data..The main purpose of “**University Management System**” project is to easily maintain the data of student in an institution.The design and development of this project provides the user the complete information of particular student like USN , Name, Branch, Semester,aadhar number .

## CHAPTER 1

# INTRODUCTION

### 1.1 Introduction to file structure

File Structures is the Organization of Data in Secondary Storage Device in such a way that minimizes the access time and the storage space. A File Structure is a combination of representations for data in files and of operations for accessing the data. File Structure allows applications to read, write and modify data. It also supports finding the data that matches some search criteria or reading through the data in some particular order. An improvement in file structure design may makes an application hundreds of time faster. The details of the representation of the data and the implementation of the operations determine the efficiency of the file structure for particular applications.

The goal of File Structure is to get the information we need with one access to the disk.

If it is not possible, then get the information with as few accesses as possible. Group information so that we are likely to get everything we need with only one trip of the disk. It is relatively easy to come up with File Structure designs that meet the general goals when the files never change. When files grow or shrink when information is added and deleted, it is much more difficult.

Goal of this course is with reference to time and space is to first minimize number of trips to the disk in order to get desired information. Ideally get what we need in one disk access or get it with as few disk accesses as possible. Secondly grouping related information so that we are likely to get everything we need with only one trip to the disk for example name, address, phone number, account balance.

Good File Structure design must have:

- Fast access to great capacity.
- Reduce the number of disk accesses
- By collecting data into buffers, blocks or buckets.



### 1.1.1 History

- History of File Structure design, In the beginning the access was sequential, and the cost of access grew in direct proportional to the size of the file, so Indexes were added to files.
- Indexes made it possible to keep a list of keys and pointers in a smaller file that could be searched more quickly.
- Simple indexes become difficult to manage for dynamic files in which the set of keys changes. Hence Tree Structures were introduced.
- Trees grew unevenly as records were added and deleted, resulting in long search requiring multiple disk accesses to find a record. Hence an elegant, self-adjusting binary tree structure called an AVL Tree was developed for data in memory.
- Even with a balanced binary tree, dozens of accesses were required to find a record in moderate sized files.
- A method was needed to keep a tree balanced when each node of the tree was not a single record as in a binary tree, but a file block containing hundreds of records. Hence B-trees were introduced.
- AVL trees grow from top down as records are added, B-trees grow from the bottom up.
- B-trees provided excellent access performance but, a file could not be accessed sequentially with efficiency.
- The above problem was solved using B+ tree which is a combination of a B-tree and a sequential linked list added at the bottom level of the B-tree.
- To further reduce the number of disk accesses, Hashing was introduced for files that do not change size greatly over time.

### 1.1.2 About File

A File is an object on a computer that stores data, information, settings, or commands used with a computer program. In a graphical user interface such as Microsoft Windows, files display as icons that relate to the program that opens the file. For example, the picture is an icon associated with adobe acrobat PDF files, if the file was on your computer, double clicking the icon in Windows would open that file in adobe acrobat or the PDF reader installed on the computer. A File is created using a software program on the computer. For example, to create a text file we use text editor, to create an image file we use an image editor, and to create a document we use a word processor. Files are not made for just reading the contents; we can also perform some operations on the Files.

- Read operation: Meant to read the information which is stored into the files
- Write operation: For inserting some new contents into a file.
- Rename or change the Name of the file.
- Copy the file from one location to the other.
- Sorting or arrange the contents of the file.
- Move or cut the file from one place to another.
- Delete a file.
- Execute Means to Run means File Display Output.

### 1.1.3 Various Storage Kind of Fields and Records

**Field Structures:** There are many ways of adding structure to files to maintain the identity of fields. Four most common methods follow:

- Method 1: Force the fields into a predictable length. Fix the length of fields. The fields of the file vary in length to make the fields fixed length we have to predict lengths.
- Method 2: Begin each field with a length indicator, store the field length just ahead of the field.
- Method 3: Place a delimiter at the end of each field to separate it from the next field.
- Method 4: Use a “keyword = Value” expression to identify each field and its contents

**Record Structures:** A record can be defined as a set of fields that belong together when the file is viewed in terms of a higher-level organization. Five most common methods follow:

**Method 1:** Make the records be a predictable number of bytes in length.

**Method 2:** Make the record be a predictable number of fields in length.

**Method 3:** Begin each record with a length indicator consisting of a count of the number of bytes that the record contains.

**Method 4:** Use a second file to keep track of the beginning byte address for each record.

**Method 5:** Place a delimiter at the end of each record to separate it from the next record.

### 1.1.4 Application of File Structure

#### **Amaze file manager:**

Amaze File Manager is a newer app comparatively speaking and it's a pretty good. It's open source and focuses on as lighter experience for those who just need to do some light file browsing. It features Material Design, SMB file sharing, a built-in app manager to uninstall apps, root explorer, and more. It manages to include the most important stuff without feeling bloated. It's free to download and use with optional in-app purchases in case you want to help fund development.

#### **Asus file manager:**

It's not every day we see an OEM app make an app list, but File Manager by ASUS is actually really good. It's compatible with most devices, even non-ASUS ones. You'll also get a clean, simple interface with LAN and SMB support, cloud storage support, support for various types of files, archiving support, and more. It's entirely free with no in-app purchases and provides a great experience for a simple file browser. About the only negative part is the lack of root access.

**ES file explorer pro:**

ES File Explorer has been around as long as most Android nerds can remember and comes with pretty much every feature you can ask for in a file browser. A while back, it was purchased by another company. Since then, things haven't gone well. The free version of the app, while very capable, now has a ton of added bloat ware that not only doesn't add to the experience, but activity subtracts from it. Thankfully, the pro version of the app doesn't have these features.

**File manager:**

File Manager is blandly named, but it's actually quite good. It's a newer file manager app that gives you one of the best sets of features without adding too much bloat. You'll get basic file management features along with cloud storage features, NAS support, and more.

**MK explorer:**

MK Explorer is another newer file manager option. It's a simple option that doesn't have a whole lot of flair. That is extremely preferable if you really just want something simple. It features a Material Design interface, the basic file management features (copy, paste, delete, SD Card support for Lollipop), and root access. There are also support for 20 languages and it has a built-in text editor, gallery, and music player. It doesn't have anything like cloud storage or network storage support, but that's not really what it's for. It's a good, cheap option.

**X-Plore file manager:**

X-Plore file Manager is one of the more unique options on the list. It's a forced dual-pane app which means you'll be managing two windows at once pretty much all the time. This is kind of cool if you're copy/pasting between folders or need to move files quickly. It also comes with support for various types of files, cloud storage, network storage, a built-in hex editor, root support and plenty of other features

## CHAPTER 2

### System Analysis

#### 2.1 Analysis of Application

The application deals with the maintenance of university, student information with in the university. This project of involved the automation of student information that can be implemented in different college managements.The main purpose of this application is to easily maintain the data of student in an institution.The design and development of this project provides the user the complete information of particular student like USN , Name, Branch, Semester.Development process of the system starts with System analysis. System analysis involves creating a formal model of the problem to be solved by understanding requirements.Application page contains following options:-

- ADD STUDENT
- SEARCH STUDENT
- UPDATE STUDENT
- DISPLAY/VIEW STUDENT DETAILS
- DELETE STUDENT
- EXIT

#### 2.2 Structure used to Store the Fields and Records

Here we are using CSV file to store the data.CSV file is very easy to maintain the data and searching of data inside the file is very efficient using variable length record

#### 2.3 Operation Performed On A File

The files are used to store the required information for its later uses. There are many file operations that can be performed by the computer system.

Here are the lists of some common file operation:

- File Create operation
- File Delete operation
- File Open operation

- File Close operation
- File Read operation
- File Write operation
- File Search operation
- File Append operation

**File Create operation:-**

The file is created with no data.

- The file create operation is the first step of the file.
- Without creating any file, there is no any operation can be performed.

**File Delete operation: -**

- File must have to be deleted when it is no longer needed just to free up the disk space.
- The file delete operation is the last step of the file.
- After deleting the file, it doesn't exist.

**File Open operation:-**

- The process must open the file before using it.

**File Close operation:-**

- The file must be closed to free up the internal table space, when all the accesses are finished and the attributes and the disk addresses are no longer needed.

**File Read operation:-**

- The file read operation is performed just to read the data that are stored in the required file.

**File Write operation:-**

- The file write operation is used to write the data to the file again, generally at the current position.

**File Search operation:-**

- The file search operation is used to search the data to the file based on primary key.

**File Append operation:-**

The file append operation is used to add records on to the end of the record.

## 2.4 Index Used

An Index is a tool for finding records in a file. It consists of Key field on which the index is searched. Reference field that tells where to find the data file record associated with particular key. Here we have used a simple indexing: We choose to organize the file as variable length record with a size field preceding each record. The fields within each record are also of variable-length but are separated by delimiters. We form a Primary key by concatenating the series id given by the admin with 'ID' at start which forms a unique identifier.

## 2.5 Compression Used

The process of reducing the size of a data file is often referred as data compression. Compression is useful because it reduces resources required to store and transmit data. Data compression is subject to a space-time complexity trade-off. In this project, compression is implemented on the genre field of the series record and vice-versa while displaying the record.[1]

## CHAPTER 3

### System Design

#### 3.1 Design of the Fields and Records

University Management System is a minimalistic graphical user interface-based application which helps users get information about students . The information includes various things such as the name,usn,branch,semester.

**Application Dashboard:** Application Dashboard has the following functionalities:

1. ADD STUDENTS DATA
2. SEARCH STUDENTS DATA
3. UPDATE STUDENTS DATA
4. DISPLAY/VIEW STUDENTS DATA
5. DELETE STUDENT DATA
6. EXIT

#### 3.2 User Interface

- The junction between a user and a computer program. An interface is a set of commands or menus through which a user communicates with a program.
- A graphical user interface is the one which helps a user graphically to accomplish the operations of the application. It works in a GUI environment built on **Tkinter framework(Python Interface)**. Tkinter allows you to develop desktop applications. It's a very good tool for GUI programming in Python



### 3.2.1 Application Dashboard

The user interface for University Management System

Fig 3.2.1 application dashboard

### 3.2.2 Insertion of a Record

The insertion of a new record of name kaushal.

NAME	BRANCH	SEM	USN	ADHAR CARD NUMBER
amith	ise	6		df308a342b16900588a076e81e8aed32
hemanth	ise	6		370a4a4165fa10346b990231ef9f80c83
pranav n	ise	6		abc7f521f89e4be0c0e6818d3d971ad5
pranav s	ise	6		7b0115a219b2fac97d200b00ca9f6ce7
rohith	ise	6		3fd051daa1ceb2453720daf25b82f58c
selva	ise	6		a0a151a9906831569357a81810d7c3f1
shashank	ise	6		4a40813e44672d39fa77a0ab631609ff6
vedanth	ise	6	120	d676ab3c91672ea8a71eb3fe54f98ed4
zakir	ise	6	150	c4fe1e1220b81f76d32aa4db48cc996b

Fig 3.2.2 insertion of record

### 3.2.3 Deletion of a Record

The deletion of record of name kaushal

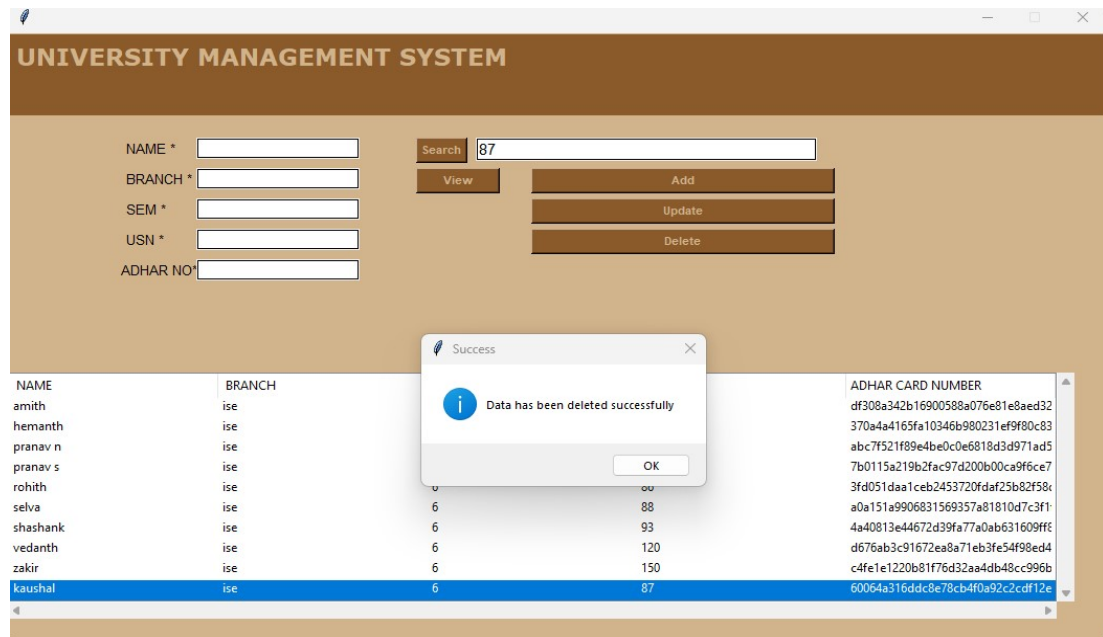


Fig 3.2.3 deletion of a record

### 3.2.4 Display of a Record

The displaying records of all students

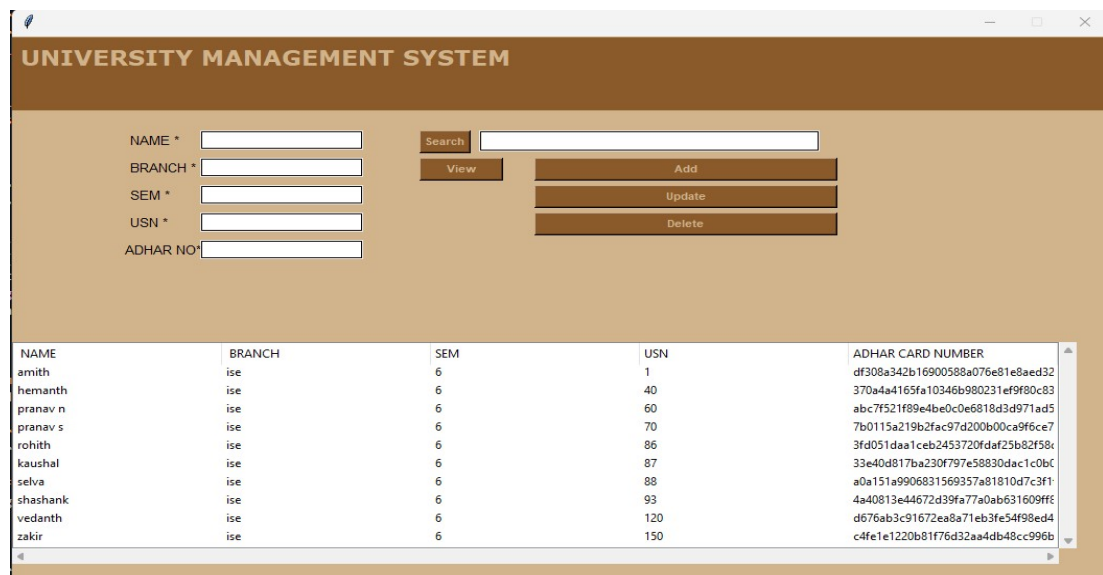


Fig 3.2.4 display of a record

### 3.2.5 Searching of a Record

The searching of new record by USN

The screenshot shows the 'UNIVERSITY MANAGEMENT SYSTEM' window. On the left, there are input fields for NAME \*, BRANCH \*, SEM \*, USN \*, and ADHAR NO\*. In the center, there is a 'Search' button and a text input field containing '87'. To the right of the search field are buttons for 'View', 'Add', 'Update', and 'Delete'. Below these elements is a table with the following data:

NAME	BRANCH	SEM	USN	ADHAR CARD NUMBER
kaushal	ise	6		1eb3f0cdf6cc2c9546153e0405f71dfff

A 'Success' dialog box is open in the center, displaying 'Data Found' with an information icon and an 'OK' button.

Fig 3.2.5 Searching of a Record

### 3.2.6 Modification of a Record

The modification of record kaushal

The screenshot shows the 'UNIVERSITY MANAGEMENT SYSTEM' window. The input fields on the left are now filled: NAME \* (S.kaushal), BRANCH \* (ise), SEM \* (6), USN \* (87), and ADHAR NO\* (354323534523423). The 'Search' button is still present with '87' in the search field. The 'View' button is now labeled 'Confirm'. The table below shows the record for 'kaushal' highlighted in blue:

NAME	BRANCH	SEM	USN	ADHAR CARD NUMBER
kaushal	ise	6		1eb3f0cdf6cc2c9546153e0405f71dfff

A 'Success' dialog box is open in the center, displaying 'data updated successfully' with an information icon and an 'OK' button.

Fig 3.2.6 Modification of a Record

### 3.2.7 Insertion of a Record Error

It displays error box message for insertion

UNIVERSITY MANAGEMENT SYSTEM

NAME \*  Search   
 BRANCH \*  View   
 SEM \*    
 USN \*    
 ADHAR NO

data

! Please fill in all fields

OK

NAME	BRANCH	USN	ADHAR CARD NUMBER
amith	ise	1	df308a342b16900588a076e81e8aed32
hemanth	ise	40	370a4a4165fa10346b980231ef9f80c83
pranav n	ise	60	abc7f521f89e4be0c0e6818d3d971ad5
pranav s	ise	70	7b0115a219b2fac97d200b00ca9f6ce7
rohith	ise	86	3fd051daa1ceb2453720daf25b82f58c
selva	ise	88	a0a151a9906831569357a81810d7c3f1
shashank	ise	93	4a40813e44672d39fa77a0ab631609ff6
vedanth	ise	120	d676ab3c91672ea8a71eb3fe54f98ed4
zakir	ise	150	c4fe1e1220b81f76d32aa4db48cc996b
kaushal	ISE	87	db4bcc21def15bc076e48e50fde6ae6

Fig 3.2.7 Insertion of a Record Error

### 3.2.8 Deletion of a Record Error

It displays error box message for deletion

UNIVERSITY MANAGEMENT SYSTEM

NAME \*  Search   
 BRANCH \*  View   
 SEM \*    
 USN \*    
 ADHAR NO

Error

✖ Select one of them from the table to delete

OK

NAME	BRANCH	USN	ADHAR CARD NUMBER
amith	ise	1	df308a342b16900588a076e81e8aed32
hemanth	ise	40	370a4a4165fa10346b980231ef9f80c83
pranav n	ise	60	abc7f521f89e4be0c0e6818d3d971ad5
pranav s	ise	70	7b0115a219b2fac97d200b00ca9f6ce7
rohith	ise	86	3fd051daa1ceb2453720daf25b82f58c
selva	ise	88	a0a151a9906831569357a81810d7c3f1
shashank	ise	93	4a40813e44672d39fa77a0ab631609ff6
vedanth	ise	120	d676ab3c91672ea8a71eb3fe54f98ed4
zakir	ise	150	c4fe1e1220b81f76d32aa4db48cc996b
kaushal	ISE	87	db4bcc21def15bc076e48e50fde6ae6

Fig 3.2.8 Deletion of a Record Error

### 3.2.9 Modification of a Record Error

It displays error box message for modification

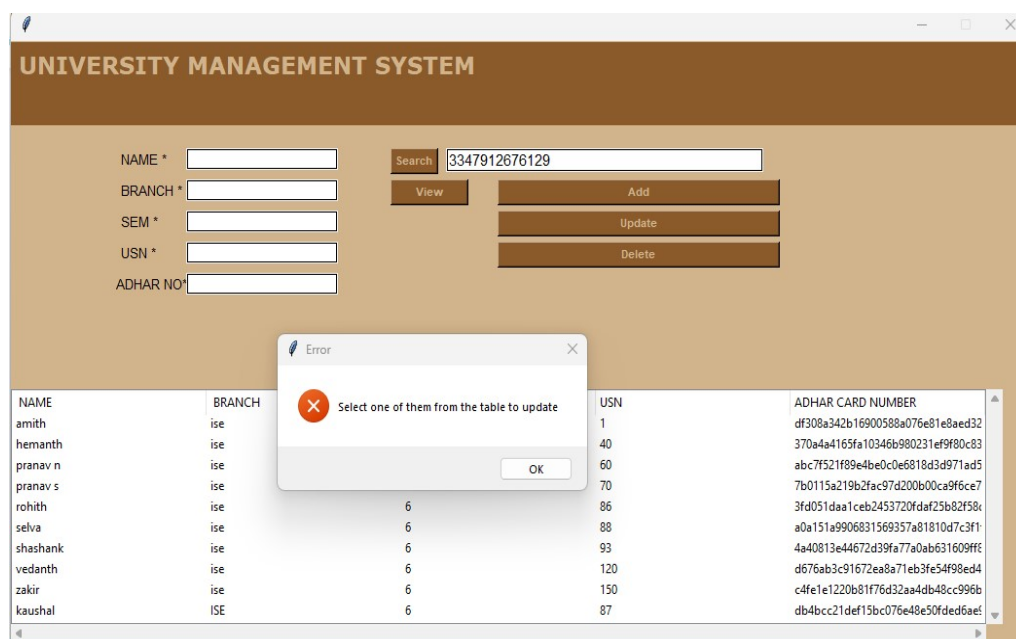


Fig 3.2.9 Modification of a Record Error

### 3.2.10 Searching of a Record Error

It displays error box message for searching

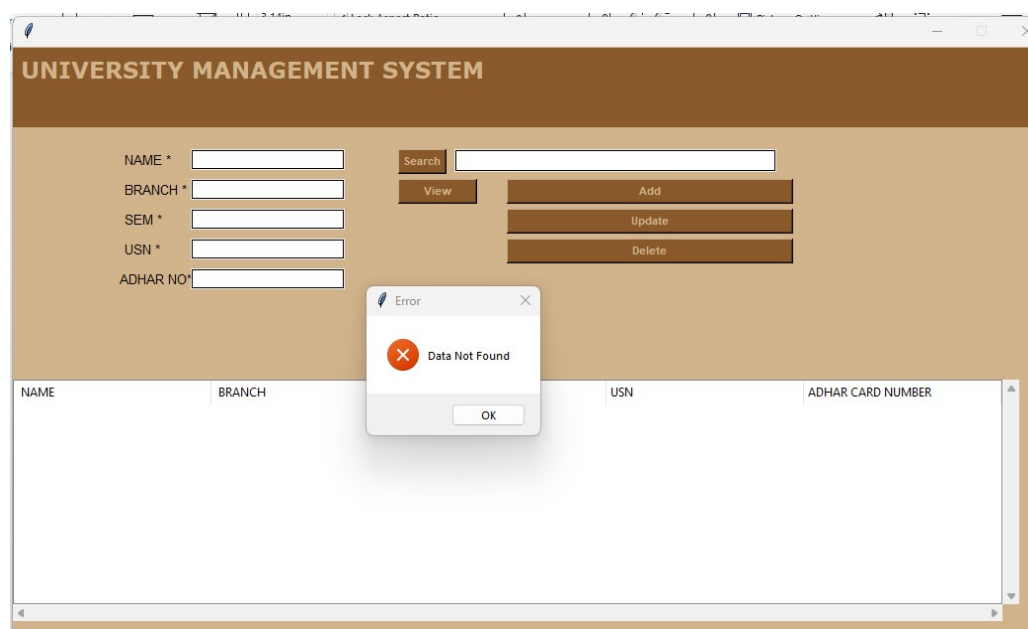


Fig 3.2.10 Searching of a Record Error

## CHAPTER 4

### **SYSTEM IMPLEMENTATION**

#### **4.1 About Python**

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language. and first released it in 1991 as Python 0.9.0. Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible

#### **4.2 Pseudo Code**

Pseudo code is a simple way of writing programming code in English. Pseudo code is not actual programming languages. It uses short phrases to write code for programs before you actually create it in a specific language. Once the functionality of the program is known, then one can use pseudo code to create statements to achieve the required results for your program.

##### **4.2.1 Insertion Operation pseudo code**

Step1 : Input the variables usn, name, branch, semester, aadhar card number for student record.

Step2 : Create file data.csv when 1<sup>st</sup> time an student details is inserted else append the details entered to data.csv file

Step3 : Create file student.csv when 1<sup>st</sup> time a student details is inserted else append the details entered to student.csv file.

#### **4.2.2 Display Operation pseudo code:**

Step1:open .csv file.

Step2:Display the data.

Step3:close the file

#### **4.2.3 Update Operation pseudo code:**

Step1:open .csv file.

Step2:Update the data.

Step3:close the file

#### **4.2.4 Delete Operation pseudo code:**

Step1:open .csv file.

Step2>Delete the data.

Step3:close the fil

#### **4.2.5 Search Operation pseudo code:**

Step1:open .csv file.

Step2:Search the data.

Step3:Display the data

Step4:close the file

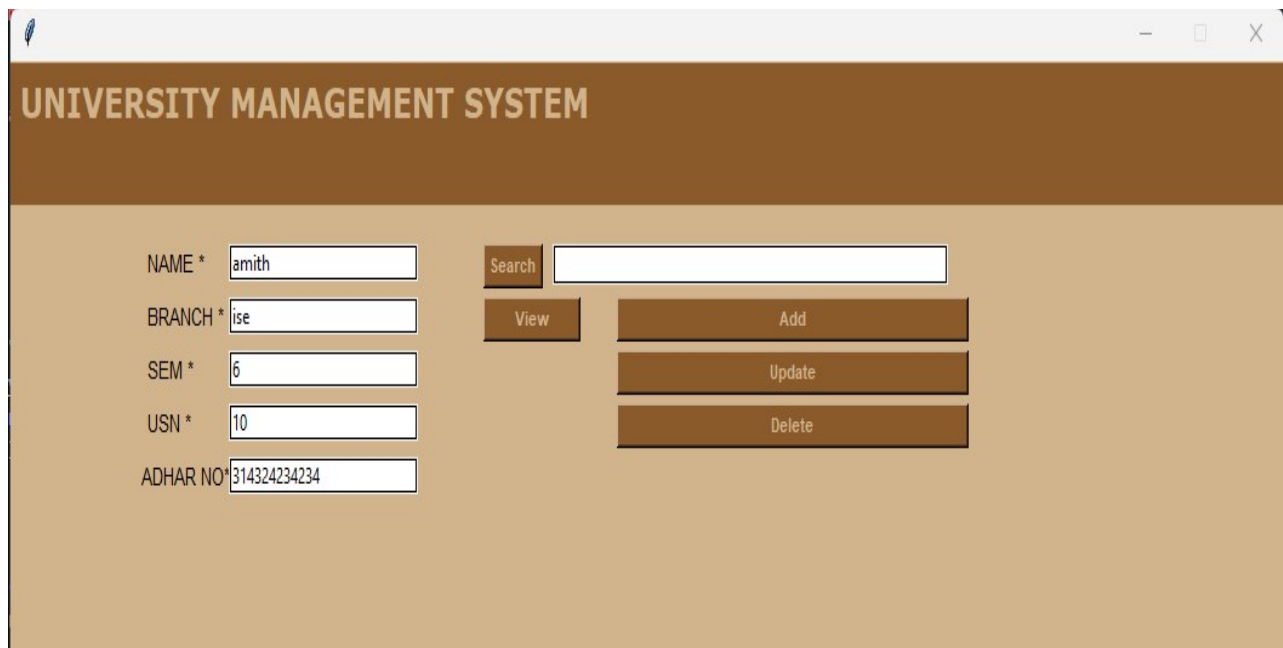
## 4.3 Testing

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do

### 4.3.1 Unit Testing

Unit testing is software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. We are going to test the components of application dashboard.-

When the user enters the data fields of the student's details such as name,branch,semester,usn and aadhar card details it should get reflect in the data.csv file.



The screenshot shows a web application titled "UNIVERSITY MANAGEMENT SYSTEM". It features a form with the following fields and buttons:

- NAME \*: amith
- BRANCH \*: ise
- SEM \*: 6
- USN \*: 10
- ADHAR NO\*: 314324234234
- Search: [empty field]
- View: [button]
- Add: [button]
- Update: [button]
- Delete: [button]

**Fig 4.3.1.1 dashboard**



```
rk2 > rk > data.csv
1 amith,ise,6,10,f45ff7f3020c5680b7089674efdc47175bf418887346f8946657ff9f93979046,$
```

**Fig 4.3.1.2 data.csv file**



### 4.3.2 Integration Testing

Integration testing is a phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation. Login and displaying the application dashboard with the top series showcase has a relation with file such as data.csv. Integration testing is done by signing a user to the dashboard and checking if the various components are working properly.

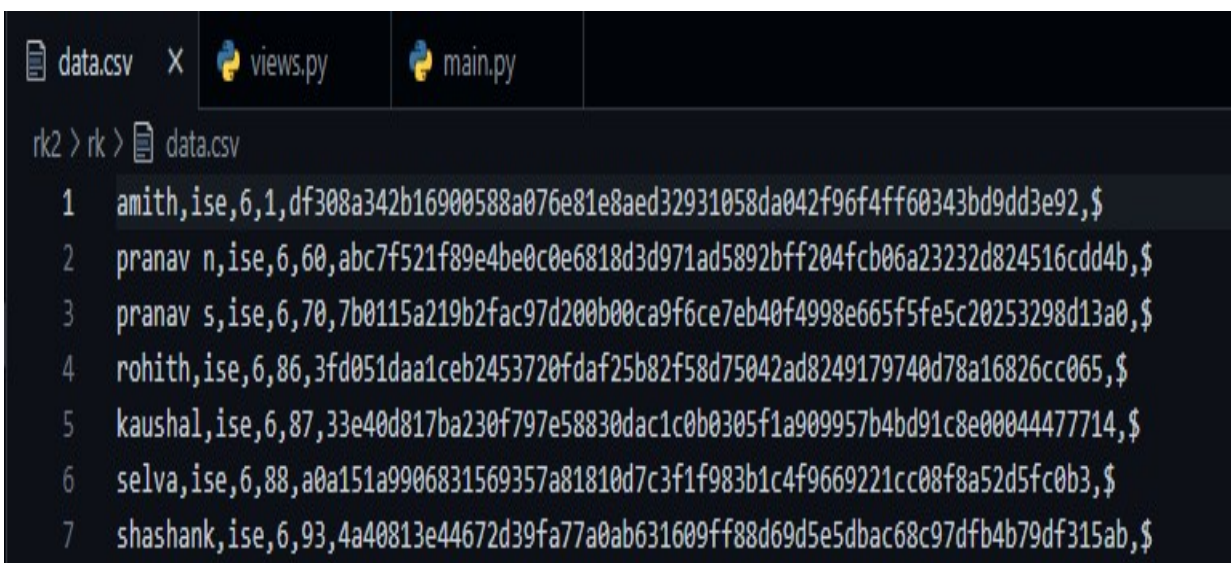
### 4.3.3 System Testing

System testing is a level of software testing where a complete and integrated software is tested. The process of testing an integrated system to verify that it needs specified requirements. This python program works well on the Windows operating system. The IDE used for coding is VS code. The code works well with Python 3.10.5 and Tkinter is required to be installed separately if newer version of Python is being used. It is recommended to run the project in a IDE that supports the latest version of the programming language. Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

## 4.4 Discussion of Results

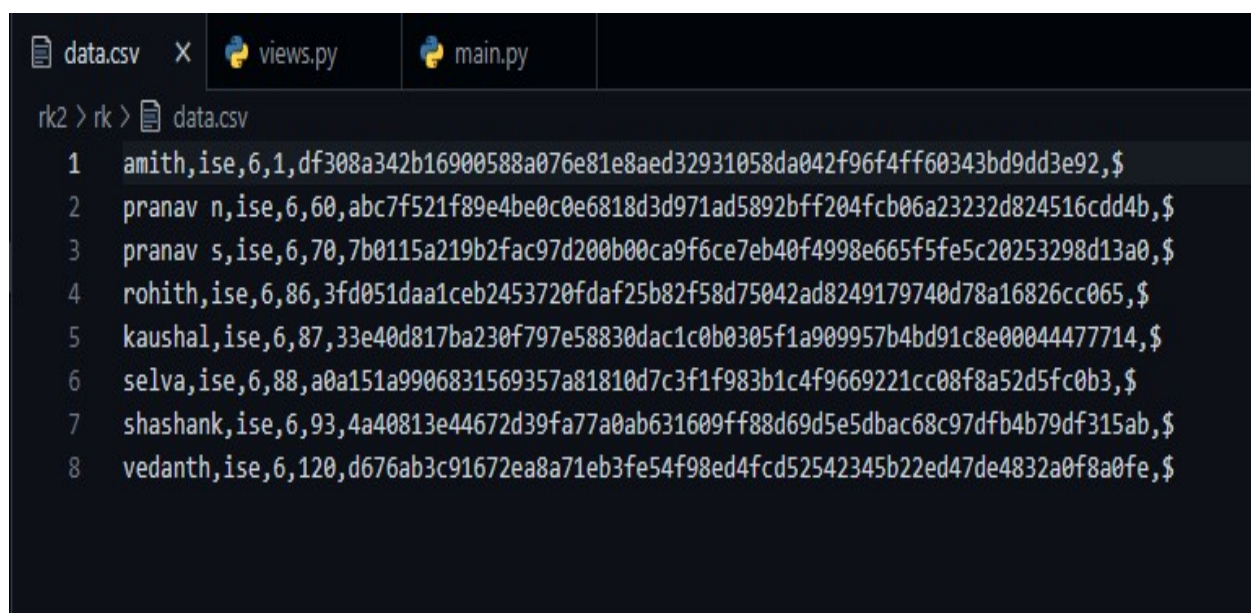
### 4.4.1 Before and After Insertion

Inserting of a new record of name vedanth after record of shashank



```
rk2 > rk > data.csv
1 amith,ise,6,1,df308a342b16900588a076e81e8aed32931058da042f96f4ff60343bd9dd3e92,$
2 pranav n,ise,6,60,abc7f521f89e4be0c0e6818d3d971ad5892bff204fcb06a23232d824516cdd4b,$
3 pranav s,ise,6,70,7b0115a219b2fac97d200b00ca9f6ce7eb40f4998e665f5fe5c20253298d13a0,$
4 rohith,ise,6,86,3fd051daa1ceb2453720fdaf25b82f58d75042ad8249179740d78a16826cc065,$
5 kaushal,ise,6,87,33e40d817ba230f797e58830dac1c0b0305f1a909957b4bd91c8e00044477714,$
6 selva,ise,6,88,a0a151a9906831569357a81810d7c3f1f983b1c4f9669221cc08f8a52d5fc0b3,$
7 shashank,ise,6,93,4a40813e44672d39fa77a0ab631609ff88d69d5e5dbac68c97dfb4b79df315ab,$
```

Fig 4.4.1 before insertion

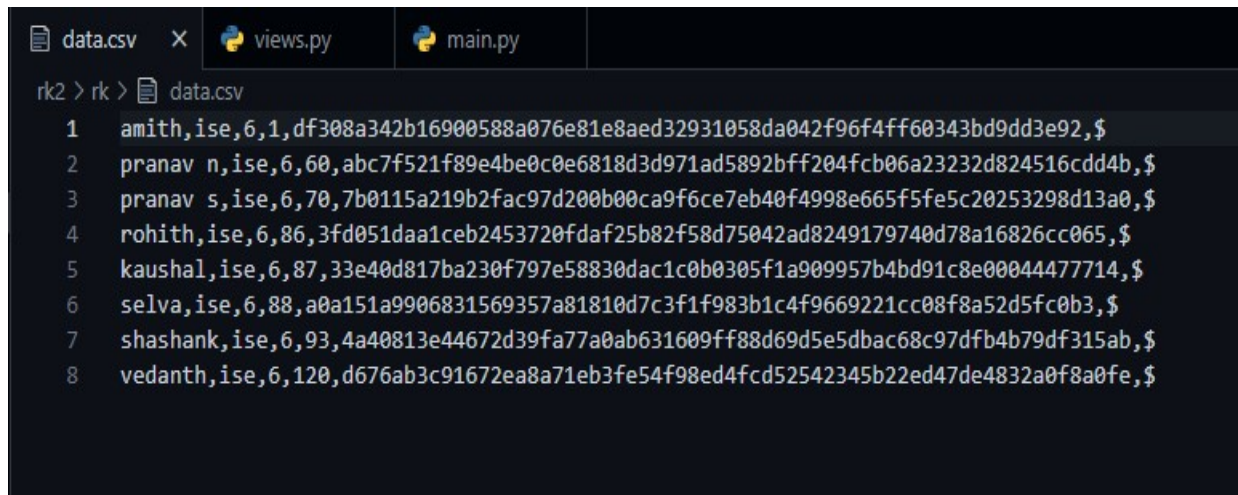


```
rk2 > rk > data.csv
1 amith,ise,6,1,df308a342b16900588a076e81e8aed32931058da042f96f4ff60343bd9dd3e92,$
2 pranav n,ise,6,60,abc7f521f89e4be0c0e6818d3d971ad5892bff204fcb06a23232d824516cdd4b,$
3 pranav s,ise,6,70,7b0115a219b2fac97d200b00ca9f6ce7eb40f4998e665f5fe5c20253298d13a0,$
4 rohith,ise,6,86,3fd051daa1ceb2453720fdaf25b82f58d75042ad8249179740d78a16826cc065,$
5 kaushal,ise,6,87,33e40d817ba230f797e58830dac1c0b0305f1a909957b4bd91c8e00044477714,$
6 selva,ise,6,88,a0a151a9906831569357a81810d7c3f1f983b1c4f9669221cc08f8a52d5fc0b3,$
7 shashank,ise,6,93,4a40813e44672d39fa77a0ab631609ff88d69d5e5dbac68c97dfb4b79df315ab,$
8 vedanth,ise,6,120,d676ab3c91672ea8a71eb3fe54f98ed4fcd52542345b22ed47de4832a0f8a0fe,$
```

Fig 4.4.1 Insertion of new record(VEDANTH)

## 4.4.2 Before and After Deletion

Deletion of existing record of name selva



```
data.csv x views.py main.py
rk2 > rk > data.csv
1  amith,ise,6,1,df308a342b16900588a076e81e8aed32931058da042f96f4ff60343bd9dd3e92,$
2  pranav n,ise,6,60,abc7f521f89e4be0c0e6818d3d971ad5892bff204fcb06a23232d824516cdd4b,$
3  pranav s,ise,6,70,7b0115a219b2fac97d200b00ca9f6ce7eb40f4998e665f5fe5c20253298d13a0,$
4  rohith,ise,6,86,3fd051daa1ceb2453720daf25b82f58d75042ad8249179740d78a16826cc065,$
5  kaushal,ise,6,87,33e40d817ba230f797e58830dac1c0b0305f1a909957b4bd91c8e00044477714,$
6  selva,ise,6,88,a0a151a9906831569357a81810d7c3f1f983b1c4f9669221cc08f8a52d5fc0b3,$
7  shashank,ise,6,93,4a40813e44672d39fa77a0ab631609ff88d69d5e5dbac68c97dfb4b79df315ab,$
8  vedanth,ise,6,120,d676ab3c91672ea8a71eb3fe54f98ed4fcd52542345b22ed47de4832a0f8a0fe,$
```

Fig 4.4.3 before deleting of record(SELVA)



```
data.csv x views.py main.py
rk2 > rk > data.csv
1  amith,ise,6,1,df308a342b16900588a076e81e8aed32931058da042f96f4ff60343bd9dd3e92,$
2  pranav n,ise,6,60,abc7f521f89e4be0c0e6818d3d971ad5892bff204fcb06a23232d824516cdd4b,$
3  pranav s,ise,6,70,7b0115a219b2fac97d200b00ca9f6ce7eb40f4998e665f5fe5c20253298d13a0,$
4  rohith,ise,6,86,3fd051daa1ceb2453720daf25b82f58d75042ad8249179740d78a16826cc065,$
5  kaushal,ise,6,87,33e40d817ba230f797e58830dac1c0b0305f1a909957b4bd91c8e00044477714,$
6  shashank,ise,6,93,4a40813e44672d39fa77a0ab631609ff88d69d5e5dbac68c97dfb4b79df315ab,$
7  vedanth,ise,6,120,d676ab3c91672ea8a71eb3fe54f98ed4fcd52542345b22ed47de4832a0f8a0fe,$
```

Fig 4.4.3 after deleting of record(SELVA)



### 4.4.3 Before and After Modification

Modification of record pranav.n to pranav hegdae

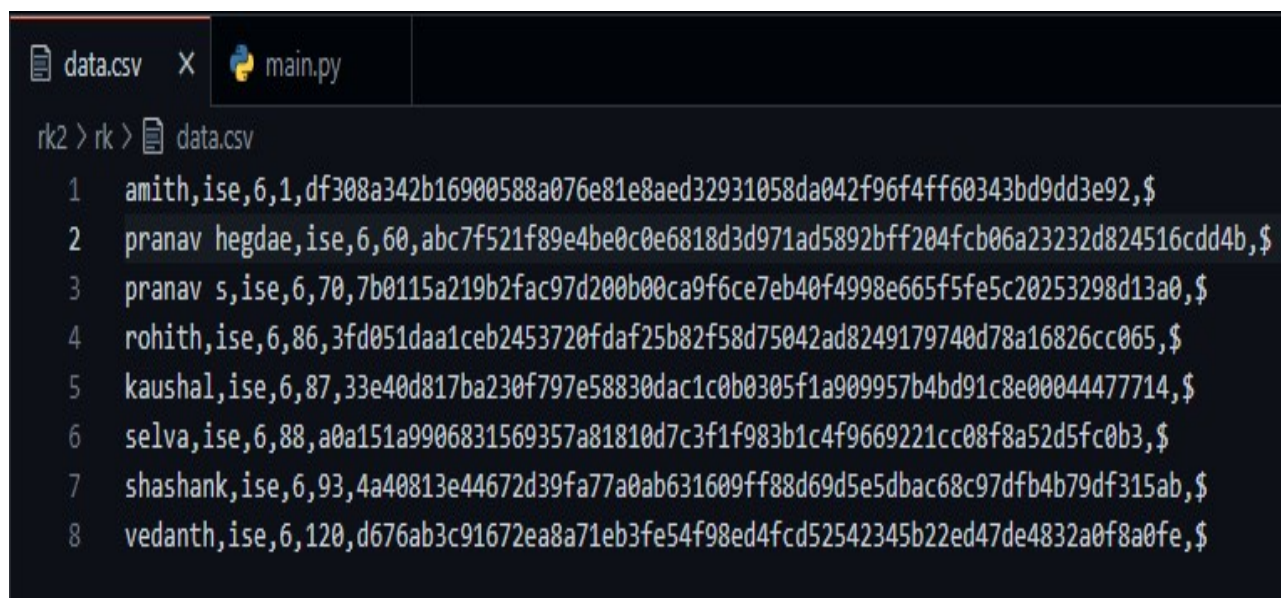


```

data.csv x views.py main.py
rk2 > rk > data.csv
1 amith,ise,6,1,df308a342b16900588a076e81e8aed32931058da042f96f4ff60343bd9dd3e92,$
2 pranav n,ise,6,60,abc7f521f89e4be0c0e6818d3d971ad5892bff204fcb06a23232d824516cdd4b,$
3 pranav s,ise,6,70,7b0115a219b2fac97d200b00ca9f6ce7eb40f4998e665f5fe5c20253298d13a0,$
4 rohith,ise,6,86,3fd051daa1ceb2453720daf25b82f58d75042ad8249179740d78a16826cc065,$
5 kaushal,ise,6,87,33e40d817ba230f797e58830dac1c0b0305f1a909957b4bd91c8e00044477714,$
6 selva,ise,6,88,a0a151a9906831569357a81810d7c3f1f983b1c4f9669221cc08f8a52d5fc0b3,$
7 shashank,ise,6,93,4a40813e44672d39fa77a0ab631609ff88d69d5e5dbac68c97dfb4b79df315ab,$
8 vedanth,ise,6,120,d676ab3c91672ea8a71eb3fe54f98ed4fcd52542345b22ed47de4832a0f8a0fe,$

```

Fig 4.4.4 before modification of record(PRANAV N)



```


data.csv x main.py
rk2 > rk > data.csv
1 amith,ise,6,1,df308a342b16900588a076e81e8aed32931058da042f96f4ff60343bd9dd3e92,$
2 pranav hegdae,ise,6,60,abc7f521f89e4be0c0e6818d3d971ad5892bff204fcb06a23232d824516cdd4b,$
3 pranav s,ise,6,70,7b0115a219b2fac97d200b00ca9f6ce7eb40f4998e665f5fe5c20253298d13a0,$
4 rohith,ise,6,86,3fd051daa1ceb2453720daf25b82f58d75042ad8249179740d78a16826cc065,$
5 kaushal,ise,6,87,33e40d817ba230f797e58830dac1c0b0305f1a909957b4bd91c8e00044477714,$
6 selva,ise,6,88,a0a151a9906831569357a81810d7c3f1f983b1c4f9669221cc08f8a52d5fc0b3,$
7 shashank,ise,6,93,4a40813e44672d39fa77a0ab631609ff88d69d5e5dbac68c97dfb4b79df315ab,$
8 vedanth,ise,6,120,d676ab3c91672ea8a71eb3fe54f98ed4fcd52542345b22ed47de4832a0f8a0fe,$

```

Fig 4.4.4 after modification of record(PRANAV HEGDAE)

#### 4.4.4 File content

The displaying of file content of records.



```
rk2 > rk > data.csv
1 amith,ise,6,1,df308a342b16900588a076e81e8aed32931058da042f96f4ff60343bd9dd3e92,$
2 pranav n,ise,6,60,abc7f521f89e4be0c0e6818d3d971ad5892bff204fcb06a23232d824516cdd4b,$
3 pranav s,ise,6,70,7b0115a219b2fac97d200b00ca9f6ce7eb40f4998e665f5fe5c20253298d13a0,$
4 rohith,ise,6,86,3fd051daa1ceb2453720daf25b82f58d75042ad8249179740d78a16826cc065,$
5 kaushal,ise,6,87,33e40d817ba230f797e58830dac1c0b0305f1a909957b4bd91c8e00044477714,$
6 selva,ise,6,88,a0a151a9906831569357a81810d7c3f1f983b1c4f9669221cc08f8a52d5fc0b3,$
7 shashank,ise,6,93,4a40813e44672d39fa77a0ab631609ff88d69d5e5dbac68c97dfb4b79df315ab,$
8 vedanth,ise,6,120,d676ab3c91672ea8a71eb3fe54f98ed4fcd52542345b22ed47de4832a0f8a0fe,$
```

Fig 4.4.5 file content

#### 4.4.5 Index File content



```
rk2 > rk > data.csv
1 amith,ise,6,1,df308a342b16900588a076e81e8aed32931058da042f96f4ff60343bd9dd3e92,$
2 pranav n,ise,6,60,abc7f521f89e4be0c0e6818d3d971ad5892bff204fcb06a23232d824516cdd4b,$
3 pranav s,ise,6,70,7b0115a219b2fac97d200b00ca9f6ce7eb40f4998e665f5fe5c20253298d13a0,$
4 rohith,ise,6,86,3fd051daa1ceb2453720daf25b82f58d75042ad8249179740d78a16826cc065,$
5 kaushal,ise,6,87,33e40d817ba230f797e58830dac1c0b0305f1a909957b4bd91c8e00044477714,$
6 selva,ise,6,88,a0a151a9906831569357a81810d7c3f1f983b1c4f9669221cc08f8a52d5fc0b3,$
7 shashank,ise,6,93,4a40813e44672d39fa77a0ab631609ff88d69d5e5dbac68c97dfb4b79df315ab,$
8 vedanth,ise,6,120,d676ab3c91672ea8a71eb3fe54f98ed4fcd52542345b22ed47de4832a0f8a0fe,$
```

Fig 4.4.6 index file content

## **Chapter 5**

### **Conclusion And Future Enhancement**

This project is developed to help the user in an institution to maintain the information of students. The main functionality of the project is to allow the user to insert, delete, search, update and display the students details in application dashboard. Furthermore, the project allow students to view their details in the application dashboard.

Future version of this project will still be much enhanced than the current version. The application currently works on non-internet environment and this can be enhanced to be compatible with the world wide web to allow greater number of users. This can be done by hosting the application on any domain. More enhancement includes Ability to store the details of various courses provided by the institution. To create login page for accessing the application dashboard. To store the details of teaching staffs..

## **REFERENCES**

- [1] [www.stackoverflow.com](http://www.stackoverflow.com)
- [2] [www.youtube.com](http://www.youtube.com)
- [3] File Structures –An Object Oriented Approach with C++ By  
MichaelJ.Folk,Bill Zoellick, Greg Riccardi
- [4] [support.office.com](http://support.office.com)
- [5 ] [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
- [6] [www.quora.com](http://www.quora.com)
- [7] [www.wikipedia.org](http://www.wikipedia.org)
- [8] [www.google.com](http://www.google.com)

