

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Machhe, Belagavi, Karnataka 590018



18ISL67 - FILE STRUCTURES LABORATORY WITH MINI-PROJECT REPORT

On

PROPERTY MANAGEMENT SYSTEM-HASHING WITH BUCKETS

*Submitted in partial fulfillment of the requirement
for the award of the degree of*

Bachelor of Engineering
in
Information Science & Engineering
by

Anish Kumar (1BG20IS008)
Piyush Singh (1BG20IS034)
Tushar Prakash (1BG20IS060)



Vidyayāmruthamashnuthe

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Information Science and Engineering

2022 – 2023

B.N.M. Institute of Technology

An Autonomous Institution under VTU

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING



Vidyayāmruthamashnuthe

CERTIFICATE

Certified that the Mini-project entitled **property management system** is carried out by **Mr. Anish Kumar** USN **1BG20IS008**, **Mr. Piyush Singh** USN **1BG20IS034**, **Mr. Tushar Prakash** USN **1BG20IS060** the bonafide student of **B.N.M Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering in Information Science & Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2022-2023. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini-project report has been approved as it satisfies the academic requirements in respect of mini-project prescribed for the said Degree.

Mrs. Laxmi V / Mrs. Kavya N L
Asst. Professor, Dept. of ISE
BNMIT

Dr. S Srividhya
Prof & Head, Dept. of ISE
BNMIT

Name & Signature of the Examiners with date:

1.

2.

List of Figures

Chapter No.	Figure No.	Description	Page No.
2	Fig.2.1	Hash Table with example	07
4	Fig.4.1	Keys loaded into has file	12
	Fig.4.2	Bucket contents	12
	Fig. 4.3	Architecture Design	13
	Fig. 4.4	Data flow diagram	14
6	Fig.6.1	Registration Page	18
	Fig.6.2	Login Page	18
	Fig.6.3	Menu Page	19
	Fig.6.4	Add Operation Page	19
	Fig.6.5	Delete Operation Page	20
	Fig.6.6	Search Operation Page	20
	Fig.6.7	Property Result	21
	Fig.6.8	Main Menu	21
	Fig.6.9	Buy Property Operation	21
	Fig.6.10	Return Property Operation	22
	Fig.6.11	User Credential Details	22
	Fig.6.12	Data save hash form	23
	Fig.6.13	Record location in hash form	23
	Fig.6.14	Borrower Relationship	24
	Fig.6.15	Simple indexing of user profile	24

List of Tables

Chapter No.	Table No.	Description	Page No.
3	3.1	Software Requirements	8
	3.2	Hardware Requirements	9

CHAPTER 1

INTRODUCTION

1.1 Objective

A property management system (PMS) is a software solution designed to help property managers and landlords streamline and automate various tasks related to managing properties. It typically includes features such as rental property listings, tenant management, lease tracking, maintenance requests, financial management, and reporting. To maintain client details line contact details, required property details, client type like residential and commercial client & price limit. To maintain property details, registration of property for sale includes property address, property description, price, facilities available.

1.2 Scope

The scope related to property management system (PMS) refers to the organization and structure of digital files and documents within the system. Here are some aspects within the scope of file structure in a PMS:

- **Property Information:** The PMS stores property-related information, such as property details, descriptions, photos, and floor plans. The file structure should include appropriate folders and subfolders to categorize and organize this data.
- **Financial Records:** The PMS handles financial transactions, so the file structure should accommodate financial records, including invoices, billing statements, and financial reports. These files should be organized in a way that enables efficient tracking and retrieval.
- **Maintenance and Repair Documentation:** Property maintenance and repair records, including work orders, service contracts, and maintenance schedules, should be structured within the file system. This allows property managers to easily locate and refer to these documents when needed.

1.3 Motivation

The motivation behind implementing a property management system (PMS) can be Revenue Optimization can help property owners and managers optimize revenue generation. With features like integrated channel management, rate management, and real-time availability updates, a PMS ensures optimal pricing, maximizes occupancy rates, and

enables effective revenue management strategies. Efficiency and Streamlined Operations is the primary motivations for adopting a PMS is to improve operational efficiency. By automating and streamlining various property management processes, such as reservations, check-ins, maintenance requests, and billing, a PMS can save time, reduce manual errors, and enhance overall productivity. Behind implementing a property management system revolves around improving efficiency, enhancing guest satisfaction, optimizing revenue, gaining data-driven insights, and ensuring compliance and security. By leveraging technology and automation, a property management system offers numerous benefits to property owners and managers, making it a valuable tool in today's competitive property management landscape.

CHAPTER 2

METHODOLOGY

2.1 Hashing with Buckets

Hashing is the process of transforming any given key or a string of characters into another value. This is usually represented by a shorter, fixed-length value or key that represents and makes it easier to find or employ the original string.

In computing, a hash table [hash map] is a data structure that provides virtually direct access to objects based on a key [a unique String or Integer]. A hash table uses a hash function to compute an index into an array of buckets or slots, from which the desired value can be found. Hashing ideally offers $O(1)$ time complexity for disk access, but the hash function can map more than one record to the same address location causing collision. Usage of buckets is one of the collision resolution technique used.

Bucket= block of records corresponding to one address in the hash table.

The hash function gives the **BUCKET ADDRESS**.

Example: for bucket holding 3 record, if we insert the following keys:

KEY	ADDRESS
JAMES	33
JORDAN	33
NIKOLA	33
KOBE	33

Fig. 2.1 Hash Table

Keys 'JAMES', 'JORDAN' and 'NIKOLA' will be placed in their address and key 'KOBE' will be an overflow record.

2.2 Tools

Language: PYTHON

Python is a popular high-level programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability with its clean and intuitive syntax, making it easier for beginners to learn and understand.

Python is an interpreted language, which means that it is executed line by line. There is no need for a separate compilation step, making development and testing Python uses dynamic typing, allowing variables to hold values of any type. Variable types are determined at runtime based on the assigned values. Python has a vibrant ecosystem with a vast collection of third-party libraries and packages. These libraries extend Python's functionality and enable developers to leverage pre-existing solutions. Python is a cross-platform language, meaning it can run on different operating systems like Windows, macOS, Linux, and more. This portability makes it easy to develop applications that can be deployed on multiple platforms.

Python's versatility, ease of use, and extensive ecosystem have contributed to its widespread adoption across various industries and domains. It is widely regarded as an excellent language for beginners and experienced developers alike, fostering rapid development and promoting code readability.

CUSTOM TKINTER

Tkinter is the standard GUI (Graphical User Interface) toolkit for Python. It provides a set of modules that allow developers to create desktop applications with graphical interfaces. Tkinter is based on the Tcl/Tk framework and provides a Python interface to it.

While Tkinter provides a rich set of GUI components and functionalities, it is also highly customizable. User can customize the appearance and behavior of Tkinter widgets by configuring their properties and options. For example, you can change the size, font, color, and alignment of text in labels, buttons, or entry fields.

With Tkinter, user can create windows, dialogs, buttons, text boxes, labels, and other GUI components. You can define event handlers to respond to user interactions, such as button clicks or menu selections. Tkinter also supports various layout managers to arrange and organize the graphical elements within windows or frames. Tkinter allows you to create custom widgets by subclassing the existing Tkinter classes and extending their functionality.

Tkinter is a mature and widely used GUI toolkit, but it may have certain limitations in terms of advanced graphical effects or modern UI designs. Tkinter allows you to create custom widgets by sub classing the existing Tkinter classes and extending their functionality. This way, you can create specialized or unique widgets tailored to your specific application's requirements.

SHA256 ENCRYPTION

SHA256 is a widely used cryptographic hash function that generates a 256-bit (32-byte) hash value, typically represented as a hexadecimal string. It belongs to the SHA-2 (Secure Hash Algorithm 2) family of hash functions. In Python, you can perform SHA256 encryption using the hashlib module, which provides an interface to various hash functions, including SHA256. Here's a detailed explanation of the steps involved in SHA256 encryption:

- Begin by importing the hashlib module, which is a built-in module in Python.
- Create an instance of the SHA256 hash object using the hashlib.sha256() constructor.
- To encrypt a message with SHA256, you need to convert it to a sequence of bytes.
- The next step is to update the hash object with the message bytes. This is done using the update() method of the hash object.
- After updating the hash object with all the message bytes, you can obtain the resulting hash digest. The hexdigest() method returns the hash digest as a hexadecimal string.

SHA256 encryption is commonly used for various cryptographic purposes, such as data integrity checks, digital signatures, password hashing, and secure storage of sensitive information. It produces a fixed-size output regardless of the size of the input message, making it suitable for hashing large amounts of data efficiently and securely.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 User Requirements

The file system is used to develop a property management system. The terms may be used in real estate, manufacturing, logistics, intellectual property, government, or hospitality accommodation management. They are computerized systems that facilitate the management of properties, personal property, and equipment, including maintenance, legalities and personnel all through a single piece of software. They replaced old-fashioned, paper-based methods that tended to be both cumbersome and inefficient. They are often deployed as client/server configurations. Today, most next-generation property management systems favour a software as a service (SaaS) model sustained by web and cloud technologies. Admin must be able to view all the table present. Admin can add a new property to the system, delete a particular property and search any property by user choice.

3.2 Software Requirements

- VS Code
- Python
- Python tkinter custom tkinter
- SHA256 encryption
- File processing with python

3.3 Hardware Requirements

- Processor: intel® core™ i5-6500 CPU @ 3.20GHz
- Installed memory (RAM): 8.00 GB (7.88 GB usable)
- System type: 64-bit operating system, x64-based processor.
- Pen and touch: no pen or touch input is available for this display.

3.4 Functional Requirements

- Whenever the admin adds a new property detail it should be inserted in the bdata.txt file.
- New record must be inserted and details are entered.
- Admin should be able to view the details of new property item inserted.
- In the display, the property details, its price and locations are displayed.
- The system must respond to user request for the operation supported by the system.

3.5 Non - Functional Requirements

- The application should display appropriate messages to the administrator if something goes wrong like inserted duplicate records
- The application should not crash.
- The user should be able to understand the facilities the application is providing to the user.
- The application should be user friendly.

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 Architectural Design

In computing, a hash table [hash map] is a data structure that provides virtually direct access to objects based on a key [a unique String or Integer]. A hash table uses a hash function to compute an index into an array of buckets or slots, from which the desired value can be found.

Hashing ideally offers $O(1)$ time complexity for disk access, but the hash function can map more than one record to the same address location causing collision. Usage of buckets is one of the collision resolution techniques used.

Consider the following set of keys, which is to be loaded into a hash file.

Key	Home Address
Green	30
Hall	30
Jenks	32
King	33
Land	33
Marx	33
Nutt	33

Fig 4.1 Keys loaded into has file

Bucket structure for the records specified above. Each bucket can hold up to three records.

Bucket contents

Green . . .	Hall . . .	
Jenks . . .		
King . . .	Land . . .	Marks . . .

(Nutt . . . is an overflow record)

Fig 4.2 Bucket contents

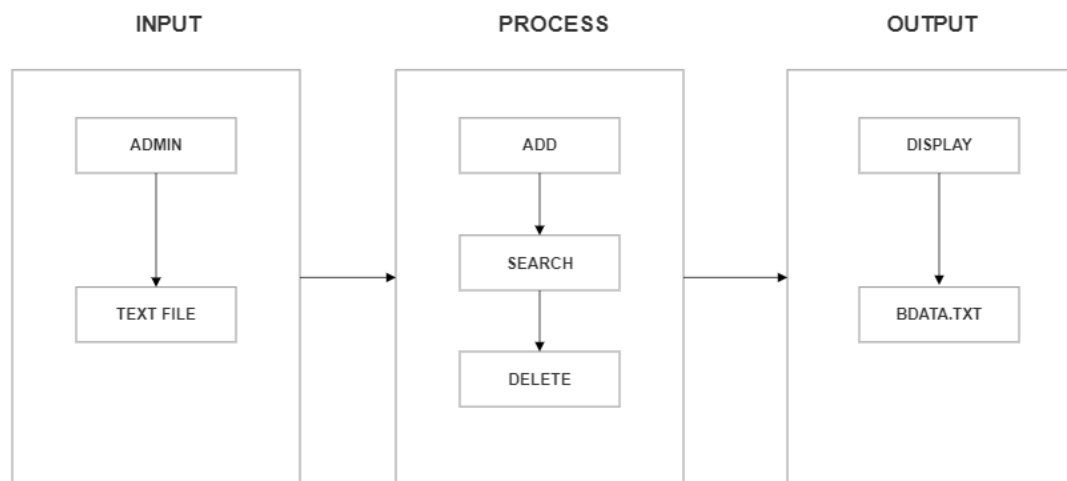


Fig 4.3 Architecture Design

In fig. 4.3, take input data through admin to fill the data will shows in the text file by the user, in the process of a data user can do the multiple modulation in the data by the add of the property, delete of the property by property id and search will also done with of by the property id by the given input through the user and the output will has been shown in display file to the user and it will basically shows in the (bdata.txt) text file will display to screen to the user by given input data will be put by the user.

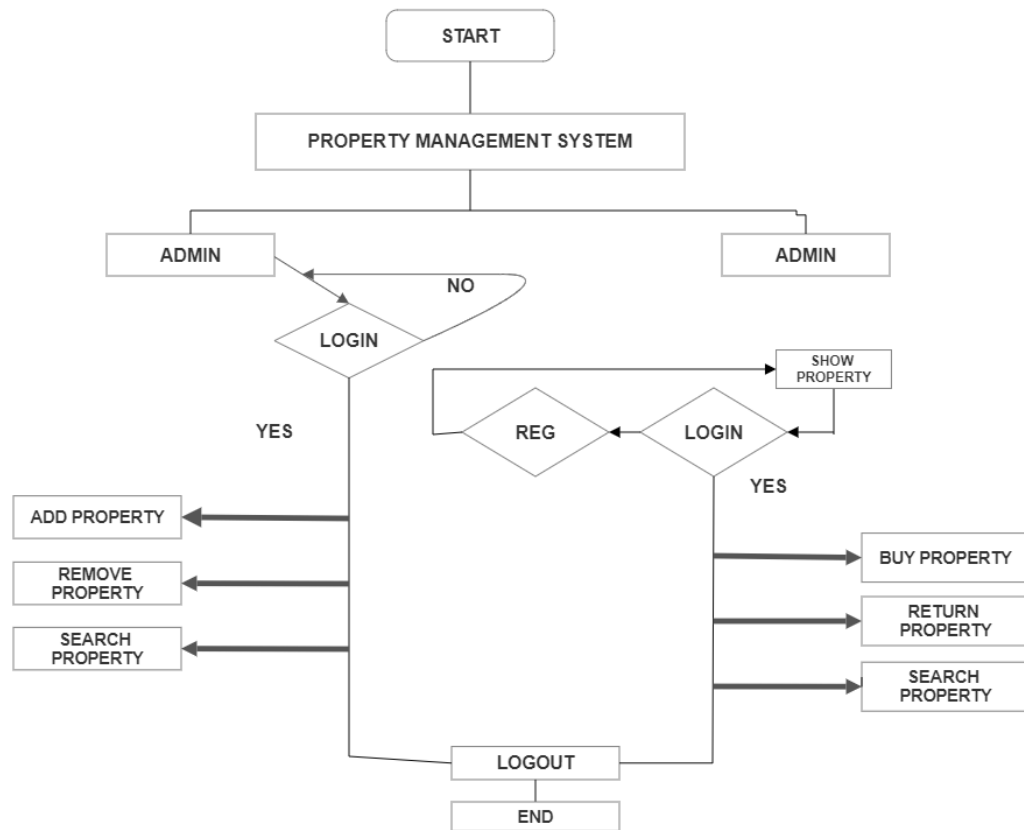


Fig 4.4 Data flow Diagram

In fig. 4.4, the admin gets an option to add property, delete property and delete the borrower id. If the admin selects add item, he has to add all the details about that particular item. When he selects the delete option, he can delete any item present in the menu. When he selects the return option, he can return any field of a particular property to that buyer id. The Buyer can sign in and place his order by selecting the desired item along with its rates. Once he adds the property details to the system, it will displayed all details of property Id after it will also search the property by property id.

CHAPTER 5

IMPLEMENTATION

5.1 Modules Implemented

Hashing with buckets is an advanced method of file organization. It generates a hash table providing the matching of records with their addresses. The hash table is generated by using a hash function.

The different operations supported by Hashing with buckets and used to implement the Property Management system are listed below:

- 1) Insert
- 2) Search
- 3) Delete

5.2 Description of Module

Insert module is used to insert details of the product to the file created by the admin. This module inserts the information in an organized manner that is pre-defined and helps in proper management of all the information. Insertion is done using the collision resolution technique-hashing with buckets.

A search function in a file structure allows you to find files or directories based on specific criteria. It involves traversing the file structure, comparing properties like file names, extensions, or content with the specified search criteria, and returning matches. The function scans through the structure recursively, locating relevant files or directories that meet the search conditions.

Delete module is used to delete a particular record that is stored in the file by using the name of that product. The admin can remove products off the menu and could remove a user as well.

5.3 Algorithm

Insert: In the insert function, hashing using buckets as collision resolution is being implemented. The item id, name, quantity and price are entered by the admin. The id is passed through a hash function which will generate a hash value. If the value is even it is inserted into bucket1 (file1.txt) and if it is odd it is inserted into bucket2 (file2.txt). If the bucket value exceeds 5, "Bucket overflow" message is displayed and finally the contents are inserted into Bdata.txt file.

1. Start
2. Details of property are entered through the frontend
3. Check if property already present
 - A) if yes show "Property already present", go to step 6
 - B) else go to step 4
4. Location in the hash table database is found using the hash function
5. Check bucket size exceeds 3
 - A) if yes "Limit Overflow" is displayed
 - B) else record is inserted in BData.txt
6. Stop

Search: A search function in hashing refers to the process of retrieving a specific element or value from a hash table or hash-based data structure. Hashing is a technique used in computer science to efficiently store and retrieve data by mapping keys to array indices using a hash function.

1. Start
2. User enters property id of the record to be searched through frontend
3. Load property record details from Bindex.txt in sorted order
4. Search property using binary search algorithm
 - A) if found return record position in hash table to step 5
 - B) else show "Property not found" and go to step 6
5. Display Property details using hash table(BData.txt)
6. Stop

Delete: Delete function aims at deleting a particular property from the main menu. The admin enters the property and that particular property id is removed from the bdata.txt file.

1. Start
2. Admin enters property id of the record to be deleted through frontend
3. Check if property is present
 - A) if yes go to step 4
 - B) else show "Property not found" go to step 6
4. Check if the property is borrowed by any customer
 - A) if yes show "Property is borrowed by some customer" go to step 6
 - B) else go to step 5
5. Property record is deleted from the hash table(BData.txt) and space is made available for reuse
6. Stop

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Snapshots of the project and description

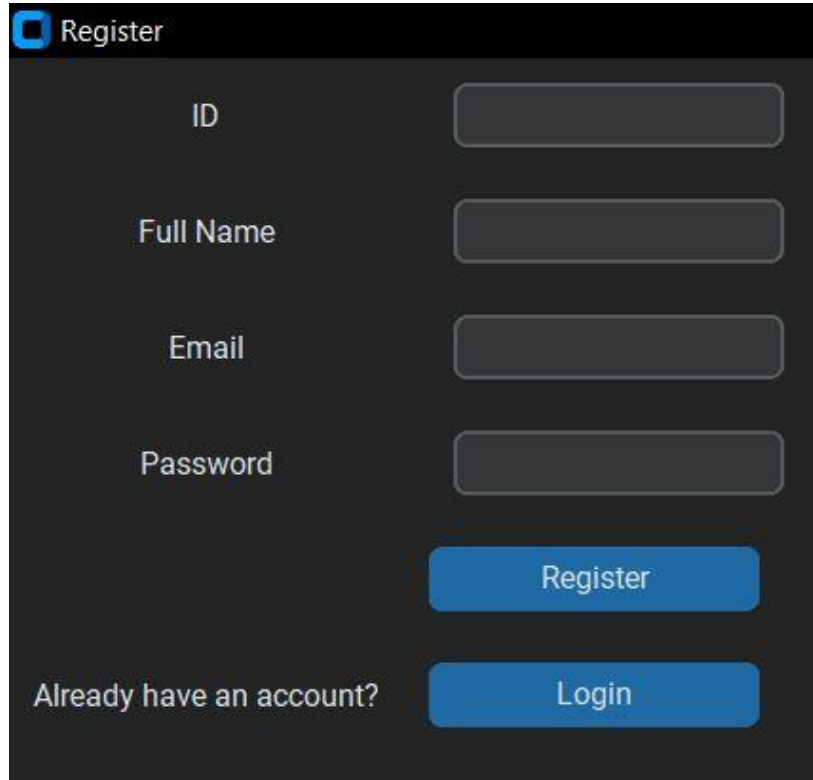
A screenshot of a registration form titled "Register" with a blue square icon. The form has a dark background. It contains four input fields: "ID", "Full Name", "Email", and "Password". Below the "Password" field is a blue "Register" button. At the bottom, there is a link "Already have an account?" followed by a blue "Login" button.

Fig.6.1 Registration Page

Fig 6.1 shows registration for the user.

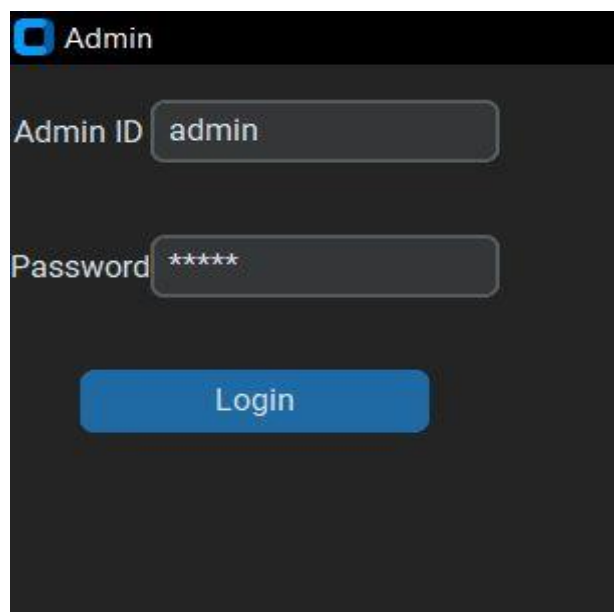
A screenshot of an admin login page titled "Admin" with a blue square icon. The form has a dark background. It contains two input fields: "Admin ID" with the text "admin" and "Password" with six asterisks "*****". Below the "Password" field is a blue "Login" button.

Fig.6.2 Admin Login Page

Fig 6.2 shows admin login for the user.

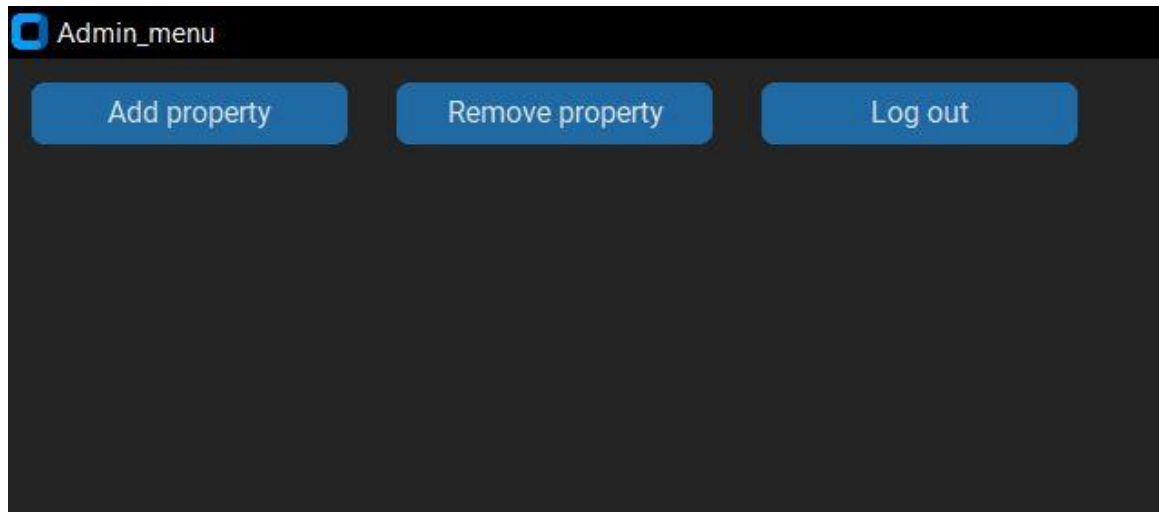


Fig.6.3 Admin MENU Page

Fig 6.3 shows admin menu page with description Modules.

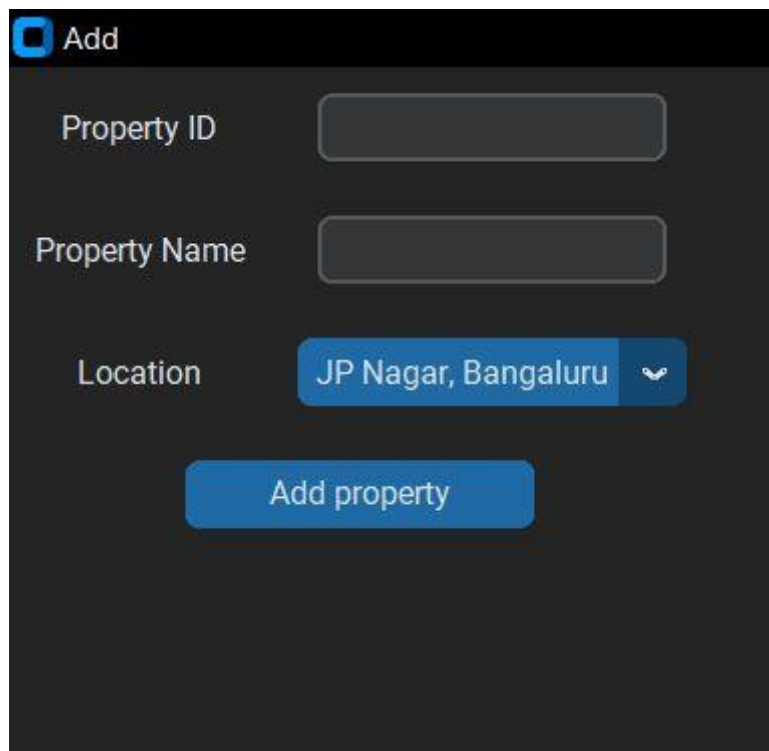
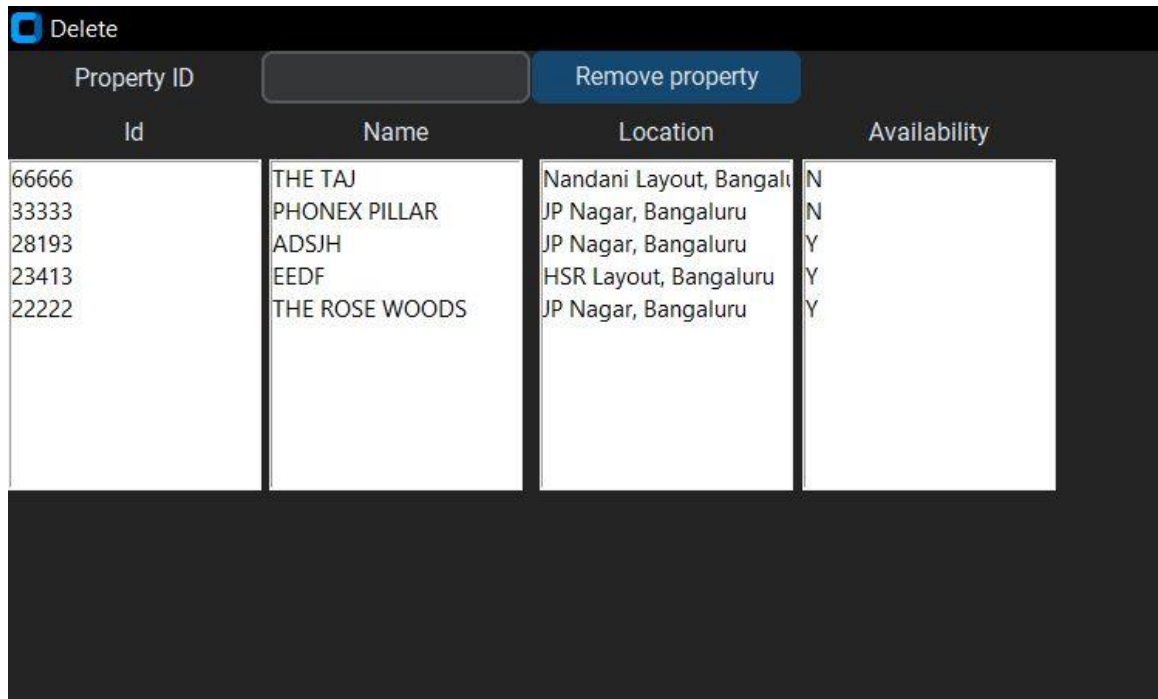
The image shows a dark-themed form titled 'Add' with a blue square icon. It contains three input fields: 'Property ID' and 'Property Name' are text inputs, and 'Location' is a dropdown menu currently showing 'JP Nagar, Bangaluru'. Below these fields is a blue rectangular button with white text that says 'Add property'.

Fig.6.4 Add Property operations

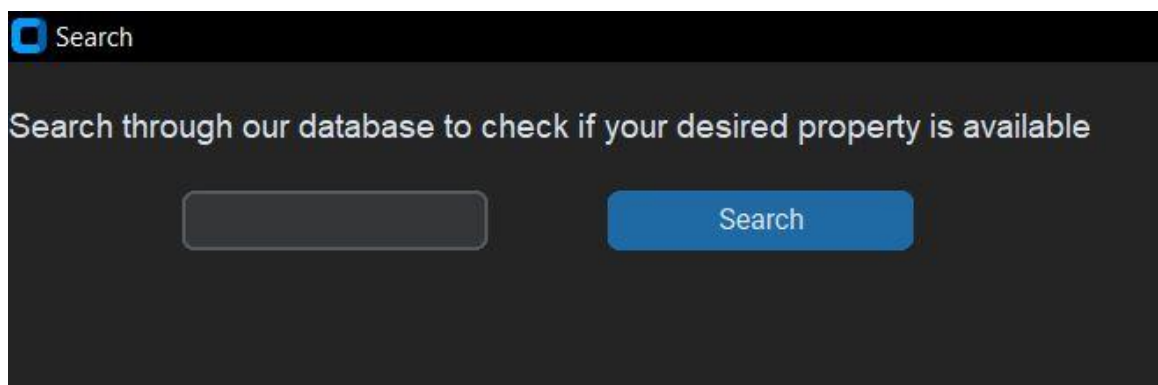
Fig 6.4 shows add property of with these details.



Id	Name	Location	Availability
66666	THE TAJ	Nandani Layout, Bangaluru	N
33333	PHONEX PILLAR	JP Nagar, Bangaluru	N
28193	ADSIH	JP Nagar, Bangaluru	Y
23413	EEDF	HSR Layout, Bangaluru	Y
22222	THE ROSE WOODS	JP Nagar, Bangaluru	Y

Fig.6.5 Deletion operations

Fig 6.5 shows Deletion property of with these details.



Search through our database to check if your desired property is available

Search

Fig.6.6 Search operations

Fig 6.6 shows Search property of with these details.

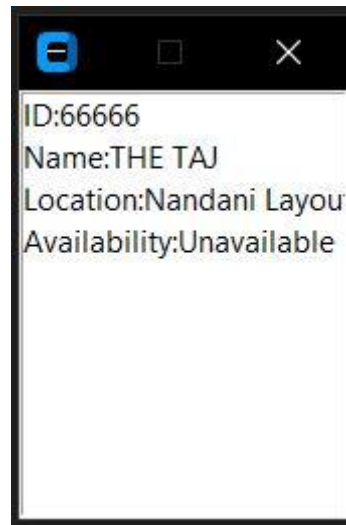


Fig.6.7 Property Search Result

Fig 6.7 shows Search property with these results.

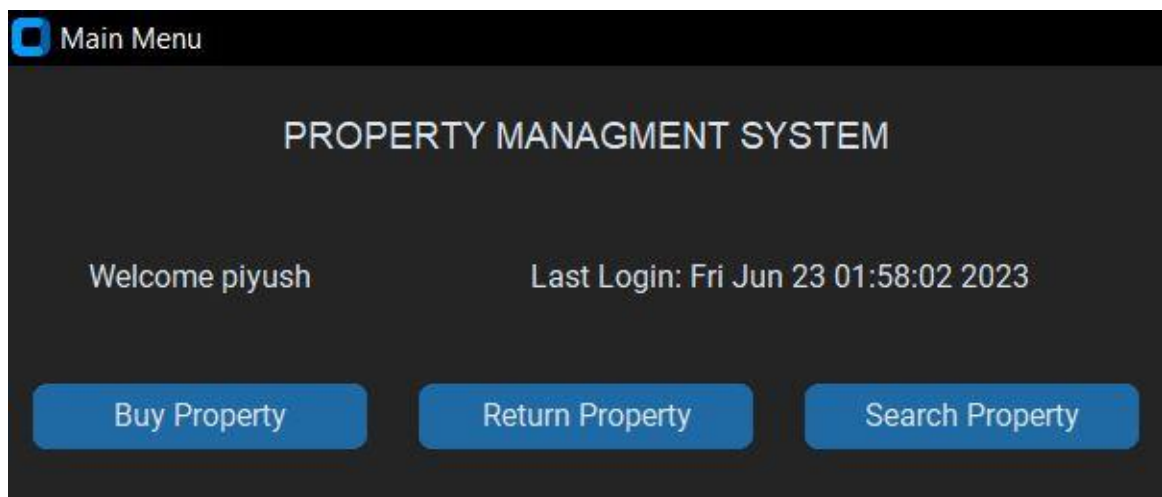


Fig.6.8 Main Menu operations

Fig 6.8 shows User property of with these details in main menu.

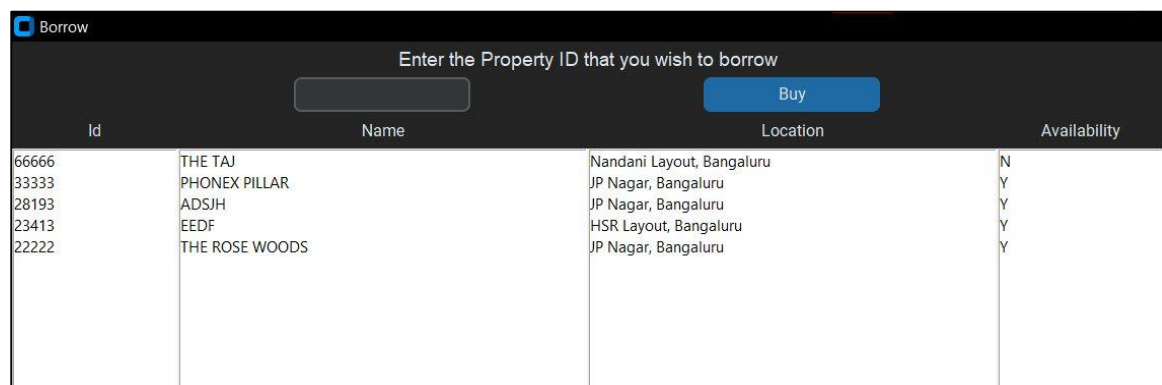
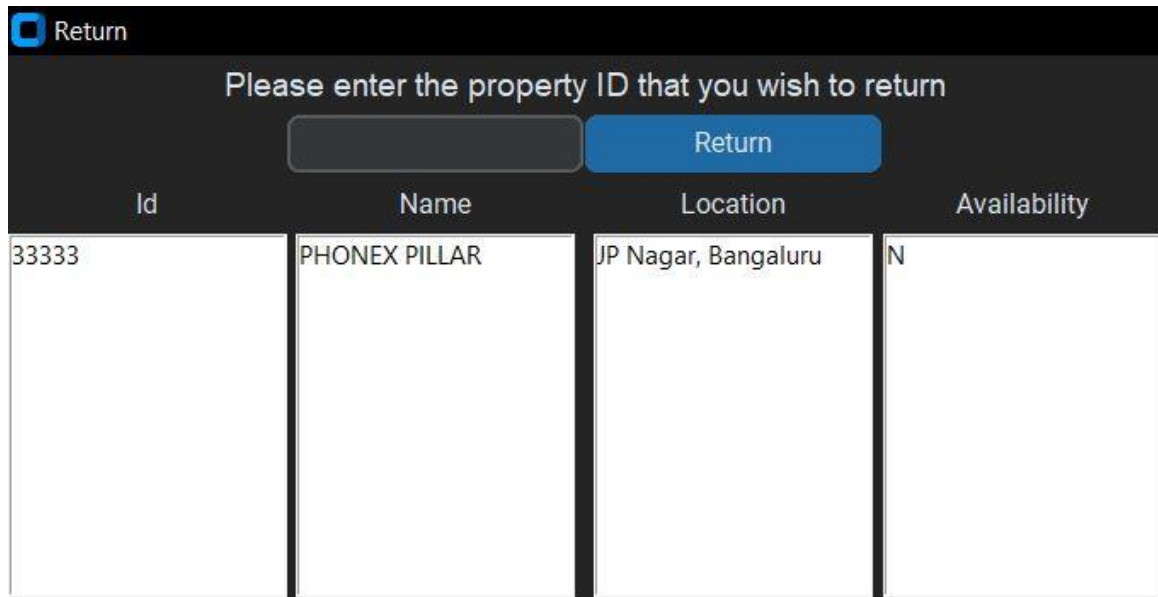


Fig.6.9 Buy Property operations

Fig 6.9 shows User to buy property with of available properties.



Return

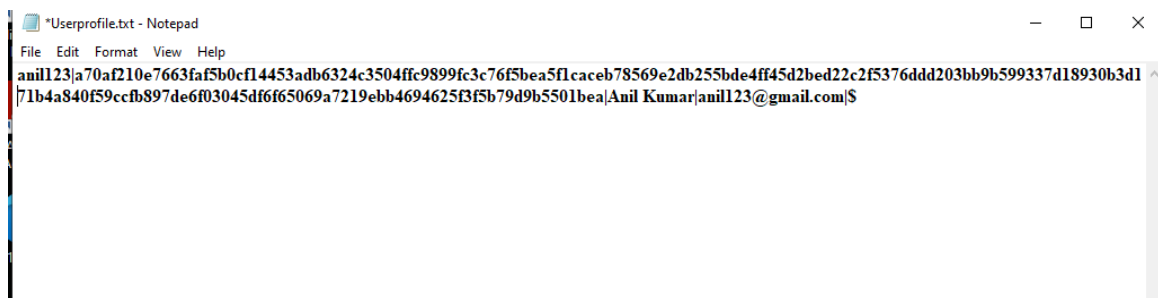
Please enter the property ID that you wish to return

Return

Id	Name	Location	Availability
33333	PHONEX PILLAR	JP Nagar, Bangaluru	N

Fig.6.10 Return Property operations

Fig 6.10 shows User to return property with of available properties.



*Userprofile.txt - Notepad

File Edit Format View Help

anil123|a70af210e7663faf5b0cf14453adb6324c3504ffe9899fc3c76f5bea5f1caceb78569e2db255bde4ff45d2bed22c2f5376ddd203bb9b599337d18930b3d171b4a840f59ccfb897de6f03045df6f65069a7219ebb4694625f3f5b79d9b5501bea|Anil Kumar|anil123@gmail.com|\$

Fig.6.11 User Credential details

Fig 6.11 shows User credential who will register the details.

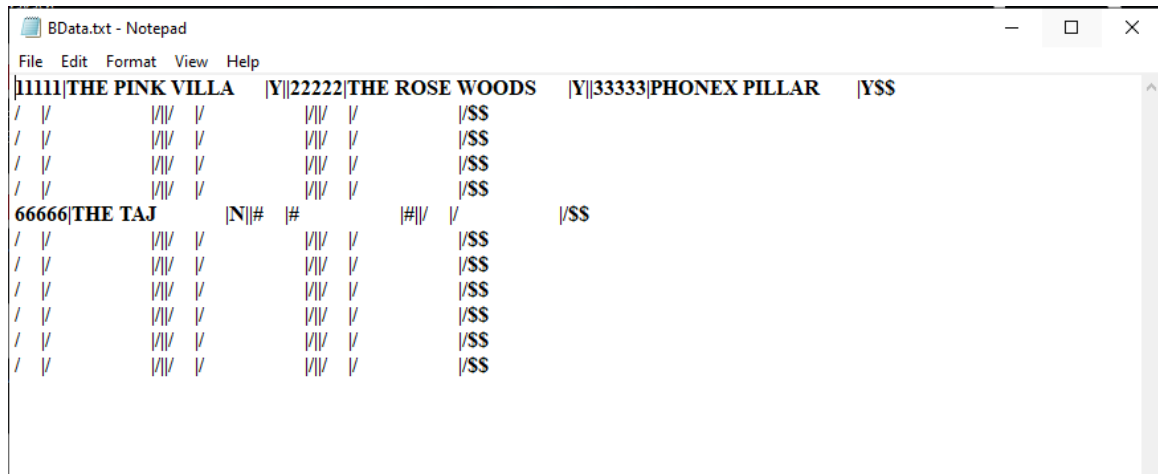


Fig.6.12 Data save in Hash form

Fig 6.12 shows admin will deleted the record and show the hash at that record.

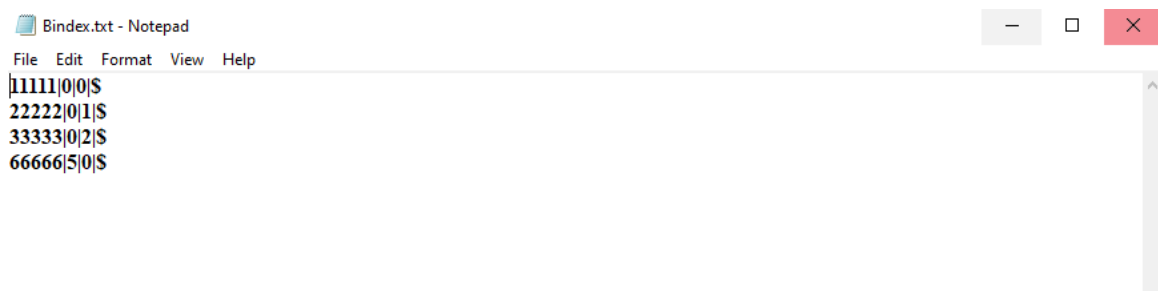


Fig.6.13 Record location Hash table form

Fig 6.13 shows admin will added the record with property id.



Fig.6.14 Borrower & property Borrowed relationship

Fig 6.14 shows in this text file that property will borrow by any user.



Fig.6.15 Simple Indexing of user profile

Fig 6.14 shows the snapshot of the content stored in "index.txt" which stores the data of the users registered in alphabetical order with the offset value of the record where the user data is stored in the file "Userprofile.txt"

6.2 Observation about the project

The project is observed to correctly perform the operations of requirements like login, add property id, search property, delete property and update property. The details when entered by the Admin for addition of properties is appended to the data file. The admin can view whole file in the form in Borrow property. Admin can add any number of properties.

Point of the project is that the file handling parts are abstracted from the user and is used. It is observed that hashing has many advantages in file structure organization. We can observe that there is no limitation of hashing for number of fields in the text file. We can insert any number of inputs to the text file. Since it resolves the collision between the inputs by using buckets. Hash function generates a hash value based on which the value goes into either bucket1 or bucket2 and is also inserted into the text file.

6.3 Advantages of hashing with buckets

The biggest advantage of hashing is that it provides the ideal disk access time. Ideally hashing provides $O(1)$ time complexity for disk access. Since the method directly maps the record to its actual address using the hash function, accessing the record would be easy by only providing the primary key to the hash function that would return the mapped address of that record. Hashing with buckets is also a collision resolution technique.

6.4 Disadvantages of hashing with buckets

The disadvantage of this method is that more than a single record could be mapped to the same address location causing collision. Hence hashing with buckets is the collision resolution method. But, by doing so we are foregoing the direct disk access provided by hashing. As in this case we'll have to traverse through the n number of buckets to find our record after obtaining the address from the hash function which is common to all buckets.

6.5 Applications of hashing with buckets

Hash buckets are used to apportion data items for sorting or lookup purposes, that it provides a lookup table mapping the required data with its address. The aim of this work is to weaken the linked lists so that searching for a specific item can be accessed within a shorter timeframe. A hash table that uses buckets is actually a combination of an array and a linked list.

CHAPTER 7

CONCLUSION

The project generalizes the file structure operations such as insert, update, search and delete, by abstracting away the core functionality of the same and to implement the optimization method Hashing using buckets as collision resolution techniques on top of the same. The project helps the user to keep track of all the information on various property id in a simple and structured file system and can view the information stored in the files by retrieval of the records. This project also enables the user to add a new entry to the file and also retrieve details from the File. Insert module allows the admin to add new items into the data.txt file. Delete module enables the admin to delete a property item from the text file and the Search module allows admin to remove the values of a particular property id.

REFERENCES

Text books

- Michael J.Folk, Bill Zoellick, File Structure, 2nd edition
- A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Englewood, N.J., Prentice Hall, 3 Edition, 1975.

Websites

- <https://www.python.org/>
- <https://www.geeksforgeeks.org/hashing-data-structure>
- <https://www.propertyangel.in/list-your-home.html>
- https://www.researchgate.net/publication/305116511_The_Property_Management_System_The_View_from_the_Front_Desk_on_Training_and_Performance