VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAM, BELAGAVI-590018

A

MINI PROJECT REPORT

ON

"HOSPITAL RECORD SYSTEM USING DATA PROCESSING AND INDEXING"

Submitted in partial fulfilment of the MINI PROJECT REPORT

In

INFORMATION SCIENCE AND ENGINEERING

VI SEMESTER

FILE STRUCTURES LABORATORY WITH MINI PROJECT (18ISL67)

By

SAJITH UMAR S	**1HK20IS086**
SYEDA ZOBIA TAZEEN Z	**1HK20IS111**

Under the guidance of

Usman Aijaz N

Assistant Professor

Department of Information Science and Engineering

2022-2023

Accredited by NAAC

HKBK COLLEGE OF ENGINEERING

22/1, Nagawara, Bengaluru – 5600 045.

E-mail: info@hkbk.edu.in, URL: www.hkbk.edu.in

HKBK COLLEGE OF ENGINEERING

Accredited by NAAC

BENGALURU – 560 045

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

A

MINI PROJECT REPORT

ON

"HOSPITAL RECORD SYSTEM USING DATA PROCESSING AND INDEXING"

Submitted in partial fulfilment of the Mini Project in

VI Semester

FILE STRUCTURES LABORATORY WITH MINI PROJECT (18ISL67)

2022-23

SUBMITTED BY

SAJITH UMAR S	**1HK20IS086**
SYEDA ZOBIA TAZEEN Z	**1HK20IS111**

DECLARATION

We hereby declare that the entire work embodied in this Project work "**HOSPITAL RECORD SYSTEM USING DATA PROCESSING AND INDEXING**" has been carried out by us during the Sixth semester of Bachelor of Engineering in Information Science and Engineering at HKBK College of Engineering, Bengaluru affiliated to Visvesvaraya Technological University, Belagavi, under the guidance of **Prof. Usman Aijaz N,** HKBK College of Engineering, Bengaluru. The work embodied in this project work is original and it has not been submitted in part time or full-time completion for anyother degree in any other university.

SAJITH UMAR S	**1HK20IS086**
SYEDA ZOBIA TAZEEN Z	**1HK20IS111**

ACKNOWLEDGEMENT

We would like to place our regards and acknowledgement to all who helped in making this project possible. There are many people who worked behind the screen to helpmake it possible the below listed are a few of them.

We would take this opportunity to express our heartfelt gratitude to **Mr. C M Ibrahim,** Chairman, **Mr. C M Faiz Mohammed**, Director and **Dr Tabassum Ara**, Principal for the entire infrastructure provided to complete the project in time.

We are deeply indebted to **Dr. A Syed Mustafa, HOD**, Information Science and Engineering for the ineffable encouragement he provided in successful completion of the project.

We sincerely thank our guide, **Prof. Usman Aijaz N** for their constant assistance, support, patience, endurance and constructive suggestions for the betterment of the project.

We are extremely thankful to the teaching and non-teaching staff of the Department of Information Science and Engineering for their valuable guidance and cooperation throughout our dissertation.

SAJITH UMAR S	**1HK20IS086**
SYEDA ZOBIA TAZEEN Z	**1HK20IS111**

ABSTRACT

Hospitals are a vital component of our life, offering the greatest medical care to those afflicted by a variety of illnesses that may be brought on by a change in the weather, an increase in work demands, etc. The hospital must keep track of its daily operations and patient and physician data in order to maintain a successful and efficient operation. However, it can be very difficult and error-prone to keep track of all the actions and their records on paper. The process takes a long time as well. Maintaining these records on paper is also neither technically feasible nor economically viable. Therefore, "Hospital Management System" is built to lessen human effort, save time, and boost the efficiency of the record.

To handle the vast number of medical data produced in hospitals, the suggested system includes cutting-edge data processing techniques. For storage space optimization and data security,it makes use of methods including data compression, encryption, and deduplication. Aside fromthat, indexing systems are put in place to allow quick and precise retrieval of patient records based on different search parameters, like patient identities, medical diagnoses, or treatment histories. Healthcare organizations can obtain streamlined record administration, decreased storage needs, greater data security, and improved accessibility to patient information by deploying this hospital record management system. Healthcare workers may quickly make educated decisions because to the system's effective data processing and indexing capabilities, which increase patient care.   .

TABLE OF CONTENTS

Chapter-1

INTRODUCTION

1.1 Introduction to file structure:

A file is a group of connected data that has a name and is stored on secondary storagesuch magnetic discs, magnetic tapes, and optical discs. Generally speaking, a file is a collection ofbits, bytes, lines, or records that have meanings determined by the user and creator of the file. The operating system's specified format should be followed when creating a file structure. Each typeof file has a specific defined structure. Text files are collections of characters arranged in lines.A source file is a list of steps and operations. An object file is a collection of data arranged into machine-understandable blocks.When an operating system specifies a different file structure, it also includes the supporting code. MS-DOS and Unix both provide a minimal number of file structures.It has to do with how you arrange your computer's files. It's a terrific approach to stay organised and make things simple for backing up your computer to keep all of your documents in one folder.An operating system also includes the supporting code when it defines an alternative file structure. Both MS-DOS and Unix offer a bare minimum of file structures.It has to do with how you organise the files on your computer. Keeping all of your documents in one folder is a great way to stay organised and make things simple while backing up your computer.

Various file structure formats are frequently employed in computer systems, including:

***Sequential File Structure*:**

In a sequential file structure, data is stored in a linear manner, with each record following the previous one. This structure is simple and easy to implement, whererecords are read or written sequentially from the beginning to the end. However, accessing or modifying specific records within the file can be time-consuming.

***Random File Structure*:**

Direct access to any record in a file is possible with a randomly generatedfile structure. The speedy retrieval and change of certain information is made possible by the unique identifier or key that is assigned to each record. Although random file formats can be moredifficult to build than sequential files, they are effective for applications that often retrieveindividual records.

***Indexed File Structure*:**

An index that maps the entries to their actual positions inside the file is part of an indexed file structure. The index normally consists of key-value pairs, where the key denotes a record's identifier and the value denotes its associated location. In contrast to sequential files, indexing enables quick retrieval of records based on their keys, facilitating speedier access.

***Hierarchical File Structure*:**

Data is arranged in a tree-like manner using a hierarchical file structure, with parent-child relationships between records. A hierarchy can be formed by the number of children each record can have. In file systems and databases, this structure is frequently used to describe the connections between data objects. Although they can be more difficult to handle, hierarchical file formats provide for quick exploration and retrieval of related items.

***Network File Structure*:**

The network file format provides relationships between records similarly to a hierarchical structure, but it permits more flexible connections. Records in this format can have numerous parent and child records, resulting in a structure that resembles a network. When working with network databases, where data items may have various relationships, this kind of file structure is frequently employed.These are but a few illustrations of file structures; in accordance with particular requirements, more modifications and combinations are also possible. The type of data, access patterns, storage efficiency, and system performance requirements all have an impact on the choice of file structure. For effective data storage, retrieval, and management within computer systems, understanding file formats is crucial.

1.2 Introduction to the project:

Our project, "Hospital Record System Using Data Processing and Indexing," aims to computerize hospital front office management and provide software that is user-friendly, straightforward, quick, and economical. It deals with gathering data about patients, doctors, receptionists, and other parties, such as diagnosis specifics. The current manual technique is cumbersome, slow, and prone to mistakes; it should be computerized for more rapid, effective results and satisfied customers. Traditionally, it was carried out by hand. The system's primary purpose is to register, store, and retrieve patient and physician information as needed, as well as to change this information in a meaningful way.

The project's goal is to use data processing and indexing techniques to create a reliable and effective hospital record management system. The maintenance of patient records is crucial in the modern healthcare sector for guaranteeing high-quality service and effective operations. Adoptingcutting-edge technologies is now necessary to simplify the record management process due to the growing amount of patient data being generated everyday.

The suggested system will make efficient use of data processing methods to manage enormous volumes of patient data. To efficiently store, retrieve, and process patient records, it will make use of advanced algorithms and data structures. The system will be able to handle sophisticated queries and produce insightful data analysis by utilizing data processing techniques. As a result, healthcare workers will be able to swiftly access pertinent patient information and make well-informed decisions, thereby improving patient care.

The system will also include indexing mechanisms to improve the accuracy and speed of data retrieval. Making data structures for indexing entails enabling effective search operations based on numerous attributes, such as patient names, medical conditions, or admission dates. The system can greatly cut down the time needed to get specific information by indexing the records, increasing the overall effectiveness of the hospital's operations. Additionally, indexing will make it simple to spot redundant or duplicate records, removing any potential data discrepancies and guaranteeing data integrity.

Data security and confidentiality will also be given top priority in the project. Protecting the data found in patient records, which includes private and sensitive information, is essential. To ensure that only authorised personnel may access and edit the records, the system will implement strong security features, such as user authentication, access limits, and encryption.

A primary focus in the development process will be adherence to pertinent data privacy laws and industry best practices. Therefore, the hospital record management system will transform how patient records are kept by utilizing data processing and indexing techniques. The system will enhance the effectiveness, accuracy, and security of record management procedures by utilizing cutting-edge algorithms, data structures, and indexing techniques. As a result, patient care will be improved, informed decision- making will be possible, and hospital healthcare services will function more effectively as a whole.

1.3 Objectives of the project

Efficient Data Processing: By automating numerous operations like data entry, storage, retrieval, and updating, the system attempts to streamline the processing of medical information. This goal is to increase efficiency while minimizing human labor and error-proneprocesses.

Centralized Record Management:

The system intends to centralize medical record storage andmanagement, making sure that all pertinent data is quickly available from a single platform.This goal contributes to greater organization and accessibility by removing the need fornumerous physical or electronic file systems.

Enhanced Data Indexing:

The system puts a lot of emphasis on using efficient indexing methods to organize and classify healthcare records. This goal permits the rapid and precise retrieval of particular records based on a variety of criteria, including patient information, a diagnosis, a course of treatment, dates, and more. It speeds up effective search processes anddecreases the time needed to find pertinent records.

Data Security and Privacy:

The system uses the proper access controls, encryption safeguards, and user authentication processes to protect the confidentiality and privacy of patient records. By maintaining compliance with data protection rules like HIPAA (HealthInsurance Portability and Accountability Act) or other relevant laws, this goal helps to safeguard private patient data from unauthorized access.

<u>Integration with Existing Systems:</u>

The system seeks integration with other hospital systems,including electronic health record (EHR) systems, laboratory information systems, billing systems, and others. This goal promotes smooth data exchange and system interoperability, which lessens duplication and boosts overall operational effectiveness.

<u>Reporting and Analytics:</u>

The system promises to offer thorough reporting and analytics capabilities, enabling hospital workers and management to learn from the data. Improvedpatient care and operational results are the result of this goal's facilitation of data-driven decision-making, trend analysis, resource planning, and performance evaluation.

<u>Scalability and Future Expansion</u>: In order to accommodate expansion and technological improvements in the future, the system aspires to be scalable and adaptive. This goal makessure the system can accommodate more capabilities or modules as needed while handling growing data volumes.

1.4 Scope of the project

The project involves designing and deploying a holistic and effective digital solution for managing hospital records. This system is designed to handle the storage, arrangement, retrieval, and administration of patient records within a medical facility. Its primary goal is to enhance operational efficiency by automating administrative and clinical procedures related to patient data. The system ensures precise data accuracy, confidentiality, accessibility, and optimized workflow. Additionally, the project entails developing a user-friendly interface that allows authorized staff members to securely input, update, and access patient information.

The project encompasses the integration of diverse modules, including electronic health records (EHR), appointment scheduling, billing and invoicing, laboratory and diagnostic results, prescription management, and medical imaging, among others. These modules are brought together into a unified system. Moreover, the scope includes the implementation of robust data security measures to safeguard patient privacy and comply with regulatory standards such as the Health Insurance Portability and Accountability Act (HIPAA). The project also involves staff training on system usage and the provision of ongoing technical support to ensure smooth operation and maintenance. The ultimate objective of the hospital record management system is to improve the overall quality of patient care, facilitate effective decision-making for healthcare professionals, reduce errors and redundancies, and enable seamless information exchange amongdifferent hospital departments.

1.5 History and evolution of technology used

1. Python:

- History: Python is a versatile and widely used programming language that was created by Guido van Rossum and first released in 1991. Its development was motivated by the desire for a language that emphasized simplicity and readability, making it accessible to both novice and experienced programmers. Python gainedpopularity due to its clean and concise syntax, extensive standard library, and strong community support. Over the years, it has evolved and adapted to meet thechanging needs of developers, incorporating new features and improvements. Python's flexibility and ease of use have made it a preferred language for a varietyof applications, ranging from web development and scientific computing to artificial intelligence and data analysis. Today, Python is one of the most widely used programming languages, with a thriving ecosystem and a vibrant communityof developers contributing to its growth and success.

- Evolution: Python has undergone significant evolution since its inception. Initially released in 1991, Python started as a simple and intuitive programming language with a focus on readability and ease of use. Over the years, it has experienced several major releases that introduced new features and improvements. Python 2, released in 2000, brought many enhancements and became widely adopted. However, as Python 2 reached its end-of-life in 2020, Python 3 emerged as the successor, addressing various design flaws and introducing backward-incompatible changes. Python 3 introduced improvements like better Unicode support, enhanced syntax, and performance optimizations. It also fostered a stronger community around the language, leading to the development of numerous libraries, frameworks, and tools. Python's versatility and extensive library ecosystem have contributed to its popularity in various domains, including web development, data analysis, machine learning, and scientific computing. In recent years, Python has continued to evolve, with regular updates introducing new features, enhancements, and performance optimizations, making it one of the most widely used and versatileprogramming languages today.

2. Command Prompt:

- History: The command prompt, also known as the command line interface (CLI), has a rich history in the world of computing. It can be traced back to the early days of computer operating systems, where text-based interfaces were the primary meansof interacting with a computer. One of the earliest command prompt implementations was the COMMAND.COM shell in MS-DOS, introduced by Microsoft in the early 1980s. This allowed users to execute commands by typing them directly into the command prompt. Similar command prompt interfaces were present in other operating systems like Unix.

- Evolution: In the early days of computing, command prompts were primarily usedby computer professionals and developers. They were often limited to a single lineof text, requiring users to type commands precisely. These prompts relied on specific commands and syntax, with minimal interactive features. As computer technology advanced, command prompts became more sophisticated. The introduction of graphical user interfaces (GUIs) in the 1980s brought a shift towards mouse-driven interactions and visual representations. However, commandprompts continued to play a vital role for power users and system administrators who required direct control and automation capabilities. the evolution of the command prompt has transformed it from a simple text

-based interface to a powerful tool with advanced features and capabilities. While graphical interfaces have become the norm for most users, the command prompt remains a valuable toolfor professionals and power users, enabling efficient system management, automation, and scripting.

3. Data Structures:

- History: For many years, computer science has investigated and used data structures. With contributions from several scholars and practitioners, numerousdata structures have been created and improved over time.
- Evolution: Tuples, Queues,Hash tables in Python and Array List are just a few of the data structures offered by Python's standard library. These data structures have been developed to provide better performance, better memory use, and more features. Additionally, third-party frameworks and libraries like Google Guava andApache Commons Collections offer specialized data structures and tools.

Performance Analysis:

- History: Software development has long been concerned with performance analysis. The performance of apps has been measured and optimized using a varietyof ways.
- Evolution: Performance analysis approaches have changed along with improvements in hardware and software. Execution time, memory use, and other performance indicators have been measured using profiling tools, benchmarking frameworks, and performance monitoring tools. These technologies have improved in sophistication and now offer thorough analysis and insights for optimization.

1.6 File Indexing History

- For a long time, file indexing has been a crucial part of information retrieval systems. The most of file indexing methods used in the early days of computing used straightforward linear procedures, which involved sequentially scanning files to create an index. However, more complex indexing methods were created to improve efficiency and search capabilities as technology evolved and the volume ofdata grew.
- with the development of data structures and algorithms, file indexing significantly improved. Tree indexing, which became a potent method for effectively storing andsearching words, was one such advancement. aims to reduce the search space and enable quick prefix-based searches by using a tree-like structure where each node represents a word's letter. This method has proven to be especially helpful in programs like word editors, spell checkers, and search engines.
- File indexing algorithms are still developing and adapting to the ever-growing volume and complexity of data in the current technological environment. Advanced indexing methods like inverted indexing and compressed indexing have become more popular as a result of the growth of massive data and the requirement for real-time information retrieval.
- Search engines frequently employ inverted indexing to effectively manage big document collections. It creates an index that associates words or keywords withthe content of the documents, enabling quick retrieval of pertinent documents in response to search queries. Ranking algorithms are frequently used in inverted indexing approaches to order the retrieved documents according to relevance.
- Compressed indexing methods concentrate on lowering the index's storage needs while keeping effective search capabilities. The index structure is compressed using a variety of compression algorithms, including delta encoding, front coding, and variable-length encoding. When dealing with big data sets or situations where storage capacity is at a premium, these strategies are very useful.
- Finally, file indexing methods have advanced significantly from the early days of linear indexing. The field has made great strides, from the advent of tree indexing and B+ tree indexing to the developments of inverted indexing and compressed indexing. File indexing will continue to develop in response to the problems of the digital era as a result of the convergence of hardware improvements and breakthroughs in algorithms and data structures. File indexing continues to be a keycomponent of effective information retrieval.

CHAPTER 2:

LITERATURE SURVEY

LITERATURE SURVEY

In a project report, a literature survey or literature review section presents an analysis of existing research and publications related to the field of interest. It examines the findings, methodologies, and outcomes of previous studies while considering the project's specific parameters and scope. This section provides a comprehensive overview of the current knowledge and understanding of the subject matter, highlighting the gaps, limitations, and potential areas for further exploration within the project. It is the most important part of your report as it gives you a direction in the area of your research.It helps you set a goal for your analysis - thus giving you your problem statement.

2.1Existing system

To save the information on patients and doctors, the hospital administration system currently keepsmanual records in the form of registers, folders, etc. Making a new appointment requires a lot of paperwork. Additionally, it causes data redundancy. The analysis phase comprises a thorough examination of the current system.

Limitations:

- Time delay Redundancy, Accuracy
- Information Retrieval.
- Storage Media

All these are rectified in the proposed system. Much emphasis is given upon the process of patient servicing.

Drawbacks associated with existing systems:

***Paper-based Documentation*:**

Many hospitals still keep patient records on paper-baseddocumentation. This can cause a number of problems, including lost or misplaced documents, trouble finding and retrieving information, and increasing storage needs.

***Limited Accessibility*:** Accessing patient records in a system that relies on paper can be laborious and time-consuming. Delays in patient treatment and decision-making may result from healthcare providers having to physically find and retrieve files.

***Data Security Risks*:** Paper documents are susceptible to loss, theft, or damage. Patient privacy and confidentiality may be jeopardized if patient information is managed improperly, lost, oreven stolen.

***Inefficient Workflow*:** Manually processing paper documents can bog down administrative procedures and make medical workflows inefficient. Staff workers may spend more time caring for patients instead of spending so much time on activities like filing, retrieving, and organizing records.

***Lack of Integration*:** Electronic health record (EHR) systems and other hospital systems are frequently not integrated into paper-based systems. This may make it more difficult for healthcare providers to work together, share data for research, and use analytics.

***Limited Data Analysis Capabilities*:** It is difficult to carry out in-depth data analysis and derive valuable insights from paper records. The manual nature of data processing makes it time- consuming and less accurate to analyze trends, spot patterns, and carry out quality improvementprogrammes.

***Higher Costs*:** Paper, printing, storage, and other physical infrastructure costs are continuing costs associated with maintaining a paper-based record management system. Additionally, the labour expenses associated with manual operations may rise as a function of how time- consuming they are.

***Difficulty in Disaster Recovery*:** Paper documents are easily lost or destroyed in emergencysituations, such as a fire or natural disaster. Critical patient information becomes extremelydifficult to recover and restore, potentially putting patient care at risk.

***Incomplete or Inaccurate Data*:** Paper records are susceptible to human error in data entry,illegible handwriting, or missing information. These mistakes may result in erroneous or incomplete patient data, which could have an impact on clinical judgement and patient safety.

2.2Proposed system

The current method uses more manpower than is necessary and has apparent flaws. The currently suggested system is fairly simple to use. The amount of work done manually will decrease when each department in the health center is computerized. The minimum amount of manpower is used.Because there is never a need to go through the files on the shelves, the patients at the registration office are registered quickly. In addition to avoiding redundant and inconsistent data, it is quick, effective, and dependable.

The adoption of an electronic health record (EHR) system is one of the main components of the suggested system. The necessity for paper-based records will be eradicated since healthcare providers will be able to keep and retrieve patient information digitally. A centralized and thoroughrepository of patient information, including medical history, test results, diagnoses, prescriptions, and treatment plans, will be made available by the EHR system. By enabling rapid and simple access to patient information, qualified healthcare providers will be able to make decisions more quickly and accurately.

The suggested system will include strong security features to guarantee data protection and privacy. Role-basedauthentication and encryption mechanisms will be used to tightly regulate access to patient records. In order to protect patient anonymity, the system would also adhere to industry standards and laws including the Health Insurance Portability and Accountability Act (HIPAA). Interoperability will be given top priority in the planned system, facilitating easy data sharing between varioushealthcare systems and providers. Improved patient care continuity and improved professional teamwork will result from this, especially when patients receive treatment from various healthcare providers.

Healthcare workers will find it simple to retrieve and update patient records because to the system's user-friendly interfaces and intuitive navigation. Additionally, it will feature sophisticated search features that let users easily find particular data in the sizable record database. Additionally, the proposed system would use data analytics and reporting tools to derive useful insights from thegathered data, which can assist healthcare organisations in identifying trends, tracking patient outcomes, and making defensible decisions to improve healthcare delivery.

Therefore, the proposed hospital record management system aims to enhance efficiency, data security, and patient care within healthcare institutions. By leveraging advanced technology and streamlined processes, the system will provide healthcare professionals with easy access to comprehensive patient records, promoting better decision-making and improved healthcare outcomes.

2.3Advantages of proposed system

***Improved Efficiency*:** The suggested solution streamlines all aspects of record management,making it possible to handle patient data more quickly and effectively. This results in less paperwork, quicker record retrieval, and enhanced efficiency throughout the institution.

***Enhanced Data Security*:** To safeguard patient records from unauthorized access or breaches, the system offers strong data security measures, such as encryption and access limits. This ensures the confidentiality of sensitive information and lowers the possibility of data loss.

***Accurate and Complete Documentation*:** By implementing the proposed system, hospitals can ensure accurate and comprehensive documentation of patient records. The system typically includes standardized templates and prompts, reducing the likelihood of missing or incomplete information.

***Easy Accessibility and Retrieval*:** The ability to quickly access and retrieve patient data is made possible by digitizing records. Authorized healthcare practitioners may rapidly look up specific records, saving time on laborious searches and facilitating swift decisions.

***Integration and Interoperability*:** Electronic health records (EHRs) and other hospital systems and equipment, as well as medical devices, are frequently able to be integrated with the proposed system. Through enabling seamless data transmission and enhancing overall healthcare coordination, this integration fosters interoperability.

***Data Analytics and Reporting*:** Through data analytics, the system can produce useful insights with the digitization of documents. Hospitals are able to do trend analysis, spot patterns, and produce reports that aid in research, quality improvement projects, and well-informed decision-making.

***Cost and Space Savings*:** By switching to a digital system instead of paper-based records, physical storage space is not required, and printing and paper costs are also decreased. The suggested solution reduces the possibility of losing or misplacing records, potentially saving the hospital money.

***Improved Communication and Collaboration*:** The system often has tools that let medical professionals interact and work together efficiently. It can enable real-time updates, secure document sharing via messaging, and effective teamwork amongst employees.

***Patient Empowerment*:** Patients can view their medical information, test results, and treatment plans through patient portals or access to electronic health records, which are frequently included in the proposed system. As a result, there is increased patient involvement, patient empowerment, and improved patient-provider communication.

***Regulatory Compliance*:** The system can help hospitals meet regulatory requirements and standards related to record keeping, privacy, and data security. It assists in adhering to guidelines such as the Health Insurance Portability and Accountability Act (HIPAA) and ensures compliance with legal obligations.

It's important to note that the advantages mentioned above may vary depending on the specific featuresand capabilities of the proposed hospital record management system.

CHAPTER 3:
SYSTEM REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENTS

1. CPU: For best performance, a recent multi-core CPU such the Intel Core i5 or anequivalent is advised. Indexing and analysis operations for the project can take advantage of the processing power provided by multi-core platforms.

2. Memory (RAM): To ensure efficient operation, the project needs at least 4 GB of RAM. Higher RAM capacity, such as 8 GB or more, may be advantageous for better performance, depending on the amount of the files being indexed and the complexity ofthe indexing techniques.

3. Storage: The project files, including the input files, index files, and any generated reports or logs, must be kept in a storage location with enough room. The size and quantity of the files being indexed determine the actual storage needs. It is advised tohave several gigabytes of unoccupied storage space.

4. Display: To view the user interface and outcomes of the project, a monitor or display screen with a resolution of at least 1024x768 pixels is required. The user experience may be improved by higher resolutions, especially when working withhuge amounts of data.

5. Input Devices: For engaging with the project's user interface and entering data,such as file selections and setup options, a normal keyboard and mouse are required.

6. Network Connectivity: Internet access is not necessary for the project's fundamental operation in terms of network connectivity. However, a reliable internet connection is required if the project calls for collecting files from distantlocations or using online resources for comparison.

3.2 SOFTWARE REQUIREMENTS

1. Python : A compatible Python version must be installed because the project was developed usingthe Python programming language. Here, we have used Python version 3.11.0.

2. Integrated Development Environment (IDE): For effective project development and debugging, an IDE like Visual studio code, Pycharm, or Pydev or Commandprompt is advised. Here, we have Command prompt also known as cmd.exe or cmd.

3. Libraries and Dependencies: Depending on the indexing methods selected and any implemented specific functionalities, the project may need extra libraries or dependencies. Utilizing build technologies like Apache Maven or Gradle, one can handle these dependencies. The project's build setup ought to contain the necessarylibraries.

3. Operating system: The project is compatible with Windows, macOS, and Linux for development and execution. It is crucial to make sure the operating system ofchoice is compatible with the Python version being utilized.

4. File System Access: Read and write access to files on the local file system arerequired for the project. Make that the person using the program has the right permissions to access the required directories and files.

5. User interface Components: Additional software components can be needed ifthe project incorporates a graphical user interface (GUI).

6. Reporting and documentation: If the project entails producing reports or documentation, software like Microsoft Office or LaTeX may be needed toefficiently prepare and format the materials.

CHAPTER:4

ANALYSIS AND DESIGN

4.1 SYSTEM DESIGN

The Hospital Record System is designed using the Python programming language. The designing be simple and can be easily understandable and operated by the person who knows the programming languages of Python.

The project contains many operations such as add, search and delete the patient details in the hospital management, so that it avoids the time consumption. The designing of the code includes file pointers which tells the current position in the file.

Define the file structure:

Create a main directory for the hospital management system. Create subdirectories within the main directory to store different types of data (e.g., patients, doctors, appointments, etc.). Each subdirectory should have a unique name that represents the type of data it contains.

Data organization:

Choose a file format to store data. Common options include CSV, JSON, or a custom file format. Determine the fields or attributes required for each type of data. For example, patient data may include fields like name, age, contact information, etc. Decide on a unique identifier for each record, such as a patient ID or an appointment ID, to ensure easy referencing and retrieval.

File operations:

Implement functions to perform basic file operations, such as creating new files, reading data from files, writing data to files, and updating existing files. Define functions to handle specific operations related to different types of data. For example, functions to add a new patient, retrieve patient details, update patient records, etc. Ensure proper error handling and validation to handle cases like file not found, incorrect data format, or missing fields.

User interface:

Develop a user interface to interact with the hospital management system. Use suitable libraries/frameworks like Flask, or Django to create a graphical or web-based interface. Connect the user interface with the file operations functions to perform CRUD (Create, Read, Update, Delete) operations on the data.

Security and access control:

Implement appropriate security measures to protect sensitive data within the system. Control access to the files and directories based on user roles and permissions. Consider encryptingsensitive data if required.

Testing and debugging:

Create test cases to ensure the system functions correctly and handles various scenarios. Debug any issues that arise during testing and ensure smooth operation.

4.2 ANALYSIS FEASIBILITY STUDY

The main objective of feasibility study is to test the technical, social and economic feasibility of developing a system. This is done before developing a system. This is done by investigating the existing system in the area under investigation and generating ideas about the new system. The system must be evaluated from the technical view point first. The assessment of the feasibility mustbe based on an outline design of the system requirement in terms of input, output. programs. procedure and staff. Having identified the outline of the system, the investigation must go on to suggest the type of the equipment, required method of developing the system and the method of running the system.
From a technical perspective, the feasibility study evaluates the hospital's existing infrastructure, hardware, and software capabilities to determine if they can support the proposed record management system. It examines factors such as compatibility, scalability, and security to ensure that the system can be effectively integrated into the hospital's existing IT framework without compromising patient data confidentiality.

I. ECONOMIC FEASIBILITY

The economic feasibility analysis focuses on the financial aspects of implementing the record management system. It involves estimating the project's cost, including software development, hardware acquisition, training, and ongoing maintenance expenses. Additionally, the study assesses the potential return on investment (ROI) by considering factors such as increased efficiency, reduced paperwork, improved patient care, and potential revenue generation.

The developing system must be justified by cost and benefit criteria to ensure that effort is concentrated on project which will give best returned the earliest. One of the factors which will affect the development of a new system is the cost it would require. Since the system is developed as a part of our study, there is no manual cost spent for the proposed system.

II. OPERATIONAL FEASIBILTY

Operational feasibility examines the practicality of implementing the system within the hospital's daily operations. It considers factors such as staff training requirements, workflow modifications, and potential resistance to change. The study also evaluates the system's impact on various departments, including medical records, billing, and clinical staff, to ensure that the implementation process does not disrupt the hospital's operations.

Proposed project would be beneficial only if they can be turned into information system that'll meet the organization operating requirements. One of the main problems faced during the development of a new system is getting acceptance from user. Being general purpose software there are no resistance from the users because this will be more beneficial to the users.

III. SCHEDULING FEASIBILITY

Scheduling feasibility involves creating a realistic timeline for the project, taking into account development, testing, and implementation phases. The study considers potential risks, dependencies, and resource availability to ensure that the project can be completed within a reasonable timeframe without adversely affecting other hospital initiatives.

In conclusion, a feasibility study of the hospital record management system is crucial to assess the project's technical, economic, operational, and scheduling aspects. By conducting a comprehensive analysis, healthcare facilities can determine the viability of implementing such a system, make informed decisions, and ensure successful project execution while enhancing patient care and operational efficiency.

IV. BEHAVIORAL FEASIBILITY

The project has been implemented by using Python and it satisfies all conditions and norms of the Organizations and the users. The proposed system "Hospital Record System" Application has much behavioral feasibility because of this the people get relaxed.

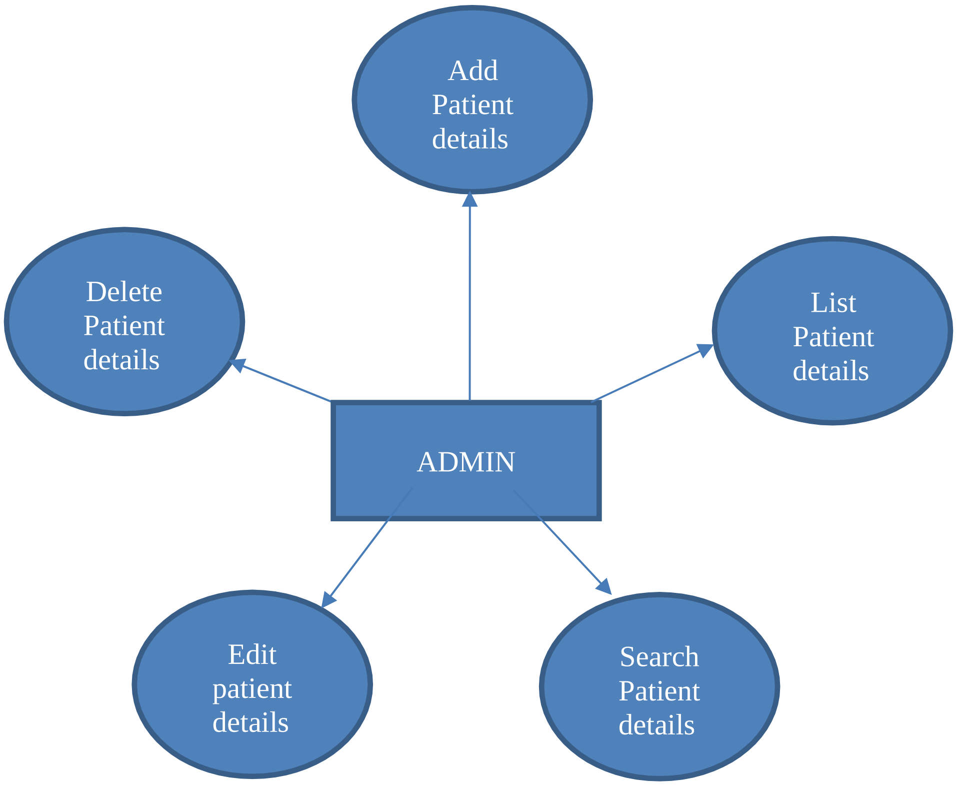


Fig : 4.1 Admin module

V. FLOWCHART OF OUR SYSTEM:

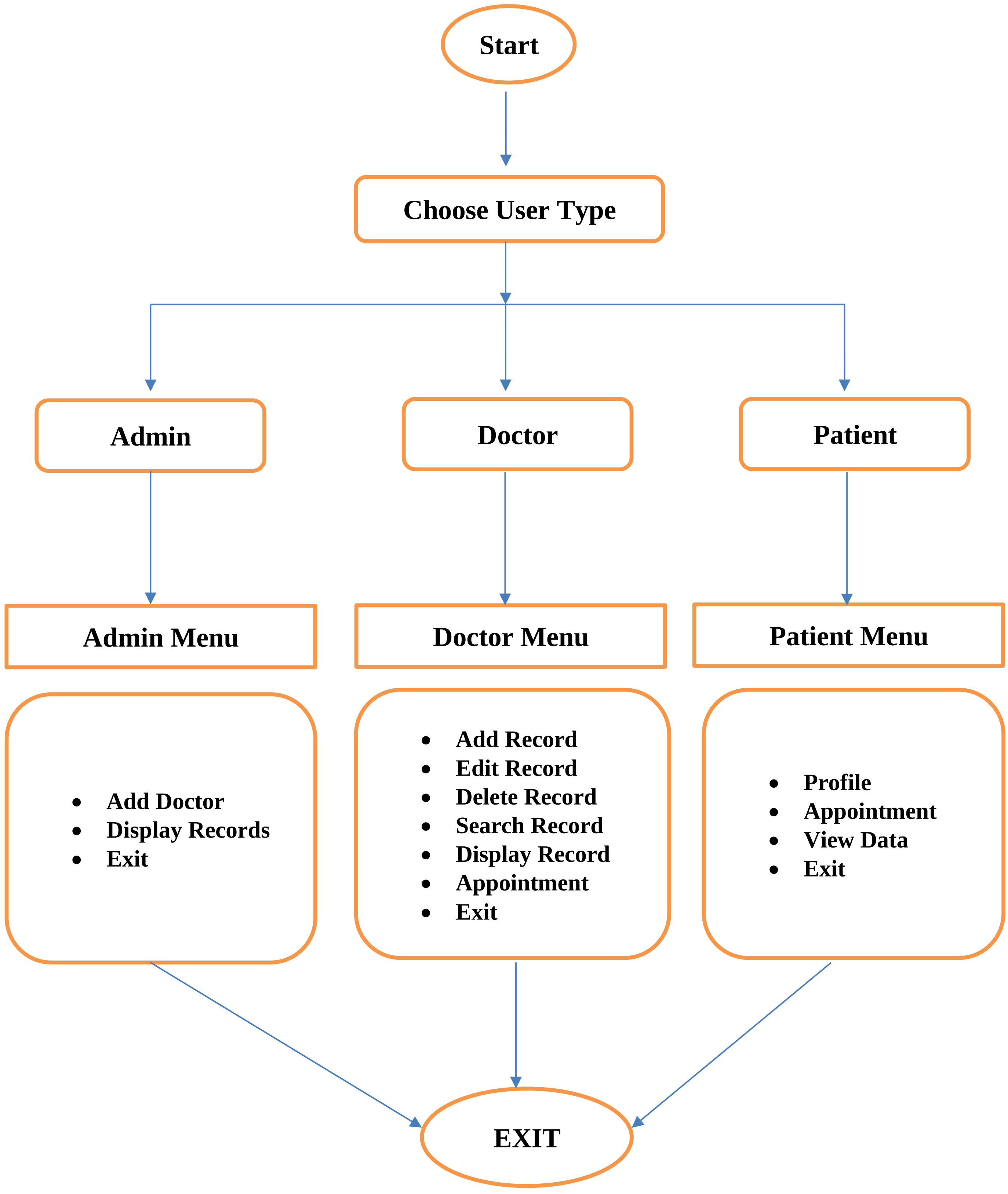


Fig:4.2 Basic system flowchart

CHAPTER:5

IMPLEMENTATION

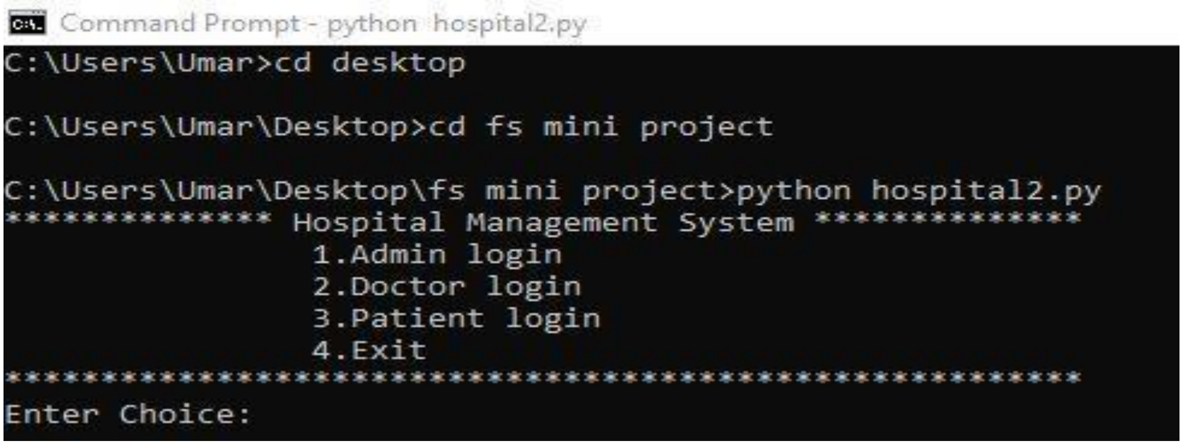
Source code:

```python
import time,base64
class admin:
    def __init__(self):
        while True:
            print('*********************************************************')
            print("""\t\t1.Add Doctor
                2.Display
                3.Exit""")
            print('*********************************************************')
            choice=input("Enter choice:")
            if choice == "1":
                self.add()
            elif choice == "2":
                self.display()
            elif choice == '3':
                break
            else:
                print('invalid choice')
    def add(self):
        print('*** Add Doctor Records ***')
        name=input("Enter Doctor name:")
        spal=input("Enter Specialization:")
        cost=input("Enter Doctor fee:")
        pas=input("Enter New Password:")
        with open("doctor.txt") as f:
            l=len(f.readlines())
            a=l+1
        docid="d00"+str(a)
        with open("doctor.txt","a") as f:
            f.write(f"{docid}|{name}|{spal}|{cost}\n")
        with open("doctor_credential.txt","a") as pf:
            new_pas=encode(pas)
            pf.write(f"{docid}|{new_pas}\n")
        print(f'added new Record Doctor ID:{docid}')
    def display(self):
        count=0
        with open("doctor.txt") as f:
            for i in f.readlines():
                a=i.split('|')
                print(f'Doctor ID:{a[0]}\nDoctor Name:{a[1]}\nDoctor
Specialization:{a[2]}\nDoctor fee{a[3]}')
                count+=1
        print(f"Total Record:{count}")
class doctor:
    def __init__(self):
        print('*********************************************************')
        self.docid=input("Enter Doctor Id:")
        passwd=input("Enter Password:")
        with open("doctor_credential.txt") as f:
            for i in f.readlines():
                id,pas=i.split('|')
                if self.docid == id and passwd == decode(pas.split("\n")[0]):
                    self.run()
                    break
            else:
                print("Not Found")
    def run(self):
        while True:
            self.ids=[]
            self.ids1=[]
            self.doc=[]
            self.data=("Patient Problem","Patient Bill")
            with open("appointment.txt") as f:
                for i in f.readlines():
                    self.ids.append(i.split("|")[0])
                    self.doc.append(i.split('|')[-1].split('\n')[0])
            with open("save.txt") as f:
                for i in f.readlines():
                    self.ids1.append(i.split("|")[0])
            print('********************************************************')
            print('''\t\t1.Add record
                2.Edit record
                3.Delete record
                4.Search record
                5.Display record
                6.Appointment
                7.Exit''')
            print('********************************************************')
            choice = int(input("Enter Choice:"))
            options={1:self.add,2:self.modify,3:self.delete,4:self.search,5:self.disp
lay,6:self.appointment}
            if choice == 7:
                break
            elif choice > 7:
                print('Invalid choice')
            elif choice < 7:
                options[choice]()
    def add(self):
        print('******* Add Patient Records *******')
        n=input('How many records to add:')
        for i in range(1,int(n)+1):
            print(f'\nEnter Patient {i} Record')
            with open('save.txt','a') as f:
                data=[]
                data1=[]
                id=input('Enter Patient Id:')
                if id not in self.ids:
                    print(f'Patient Id {id} do not have new appointment')
                    break
                else:
                    with open('patient.txt') as pf:
                        for i in pf.readlines():
                            if id == i.split('|')[0]:
                                for j in i.split('|'):
                                    data.append(j.split('\n')[0])
                    with open("appointment.txt") as rf:
                        for i in rf.readlines():
                            if id == i.split("|")[0]:
                                continue
                            else:
                                data1.append(i)
                    with open("appointment.txt","w") as wf:
                        for i in data1:
                            wf.write(i)
                    for i in self.data:
                        data.append(input(f"Enter {i}:"))
                    data.append(self.docid)
                    f.write(f"{'|'.join(data)}\n")
                    print('Added Successfully')
    def modify(self):
        id=input('Enter Patient Id:')
        modify_value=[]
        data=[]
        if id in self.ids1:
            with open('save.txt') as f:
                for i in f.readlines():
                        if id == i.split('|')[0]:
                            with open('modify.txt','a') as mf:
                                mf.write(i)
                        else:
                            modify_value.append(i)
            with open('patient.txt') as pf:
                for i in pf.readlines():
                    if id == i.split('|')[0]:
                        for j in i.split('|'):
                            data.append(j.split('\n')[0])
            for i in self.data:
                data.append(input(f"Enter {i}:"))
            data.append(self.docid)
            modify_value.append(f"{'|'.join(data)}\n")
            with open('save.txt','w') as f:
                modify_value.sort()
                for i in modify_value:
                    f.write(i)
            print("Successfully Modified")
        else:
            print('Patient Id not found!')
    def delete(self):
        saves=[]
        count=0
        id=input('Enter Patient Id:')
        if id in self.ids1:
            with open('save.txt') as sf:
                for i in sf.readlines():
                    if self.docid == i.split('|')[-1].split('\n')[0]:
                        with open('delete.txt','a') as df:
                            df.write(i)
                            count+=1
                        continue
                    else:
                        saves.append(i)
            with open('save.txt','w') as f:
                f.writelines("")
            with open('save.txt','a') as f:
                for i in saves:
                    f.write(i)
            if count <= 0:
                print('Patient Id not found')
            elif count > 0:
                print(f"Deleted Patient Id:{id}")
        else:
            print(f'Patient Id {id} not found!')
    def search(self):
        id=input('Enter Patient Id:')
        with open('save.txt') as f:
                for i,j in enumerate(f):
                    v=j.split('|')
                    if v[0] == id and self.docid == v[-1].split('\n')[0]:
                        print(f'Patient Id:{v[0]}\nPatient Name:{v[1]}\nPatient
Age:{v[2]}')
                        print(f"Patient Gender:{v[3]}\nPatient
Contact:{v[4]}\nPatient Email:{v[5]}")
                        print(f"Patient Address:{v[6]}\nPatient
Problem:{v[7]}\nBill:{v[8]}")
                        break
                else:
                    print('Patient Id not found!')
    def display(self):
        count=0
        with open('save.txt') as df:
            for i in df.readlines():
                if self.docid == i.split("|")[-1].split("\n")[0]:
                    v=i.split('|')
                    print(f'Patient Id:{v[0]}\nPatient Name:{v[1]}\nPatient
Age:{v[2]}')
                    print(f"Patient Gender:{v[3]}\nPatient Contact:{v[4]}\nPatient
Email:{v[5]}")
                    print(f"Patient Address:{v[6]}\nPatient
Problem:{v[7]}\nBill:{v[8]}\n")
                    count+=1
            print(f'Total Record:{count}')
            if count == 0:
                print("No Record!")
    def appointment(self):
        with open("appointment.txt") as af:
            for i in af.readlines():
                if self.docid == i.split('|')[1]:
                    v=i.split('|')
                    print(f'Patient Id:{v[0]}\nDoctor Id:{v[1]}\nAppointment
Date:{v[2]}\nPatient Problem:{v[3]}\n')
                    break
            else:
                print('No Appointment')
```

```
class patient:
    def __init__(self):
        while True:
            print('********************************************************')
            print('''\t\t1.Login
                2.Register
                3.Exit''')
            print('********************************************************')
            choice = input("Enter choice:")
            if choice == "1":
                self.login()
            elif choice == "2":
                self.register()
            elif choice == "3":
                break
            else:
                print("Invalid choice!")
    def login(self):
        self.id=input("Enter Id:")
        passw=input("Enter Password:")
        with open("patient_credential.txt") as f:
            for i in f.readlines():
                a,b=i.split('|')
                if self.id == a and passw == decode(b.split('\n')[0]):
                    self.run()
            else:
                print('failed')
    def register(self):
        save_data=[]
        data=("Name","Age","Gender","Contact","Email","Address")
        with open("patient.txt") as fr:
            l=len(fr.readlines())
            save_data.append(f"{l+1:0>5}")
        print('********************************************************')
        for i in data:
            save_data.append(input(f"Enter {i}:"))
        passw=input("Enter Password:")
        print('********************************************************')
        with open("patient.txt","a") as f:
            f.write(f"{'|'.join(save_data)}\n")
        with open("patient_credential.txt","a") as f1:
            passw1=encode(passw)
            f1.write(f"{save_data[0]}|{passw1}\n")
        print(f"Registered your Id is {save_data[0]}")
    def run(self):
        while True:
            print('********************************************************')
            print("""\t\t1.Profile
                2.Appointment
                3.View Doctor's
                4.Exit""")
            print('********************************************************')
            choice = input("Enter choice:")
            options={'1':self.profile,'2':self.appointment,'3':self.view}
            if choice == '4':
                break
            elif int(choice) > 4:
                print('Invalid choice')
            options[choice]()
    def profile(self):
        with open('patient.txt') as pf:
            for i in pf.readlines():
                if self.id == i.split('|')[0]:
                    v=i.split('|')
                    print(f'Patient Id:{v[0]}\nPatient Name:{v[1]}\nPatient 
Age:{v[2]}')
                    print(f"Patient Gender:{v[3]}\nPatient Contact:{v[4]}\nPatient 
Email:{v[5]}\nPatient Address:{v[6]}")
    def appointment(self):
        while True:
            print('********************************************************')
            print("""\t\t1.Take Appointment
                2.View Appointment History
                3.Exit""")
            print('********************************************************')
            choice = input("Enter choice:")
            if choice == '1':
                dId=input("Enter Doctor Id:")
                date=input("Enter Appointment Date:")
                problem=input("Enter Problem:")
                print('********************************************************')
                with open("appointment.txt",'a') as af:
                    af.write(f"{self.id}|{dId}|{date}|{problem}\n")
                with open('appointment_history.txt','a') as af1:
                    af1.write(f"{self.id}|{dId}|{date}|{problem}\n")
                print('Appointment Added Successfully')
            elif choice == '2':
                count=0
                with open("appointment_history.txt") as af2:
                    a=af2.readlines()
                    for i in a:
                        if self.id == i.split('|')[0]:
                            print(i)
                            count+=1
                if count == 0:
                    print("No Record")
                else:
                    print(f'Total Record:{count}')
            elif choice == '3':
                break
            else:
                print('Invalid choice')
    def view(self):
        with open("doctor.txt") as df:
            print(df.read().center(10))
def encode(pas):
    try:
        p1=pas.encode('ascii')
        p2=base64.b64encode(p1)
        p3=p2.decode('ascii')
        return p3
    except:
        print('Error while Encode')
def decode(pas):
    try:
        p1=pas.encode('ascii')
        p2=base64.b64decode(p1)
        p3=p2.decode('ascii')
        return p3
    except:
        print('Error while decode')
if __name__=='__main__':
    while True:
        print('************** Hospital Management System **************')
        print('''\t\t1.Admin login
                2.Doctor login
                3.Patient login
                4.Exit''')
        print('********************************************************')
        choice = input("Enter Choice:")
        if choice == "1":
            if input("Enter username:") =="admin" and input("Enter password:") ==
"admin":
                admin()
            else:
                print("invalid")
        elif choice == "2":
            doctor()
        elif choice == "3":
            patient()
        elif choice == "4":
            break
        else:
            print('Enter valid choice!')
```

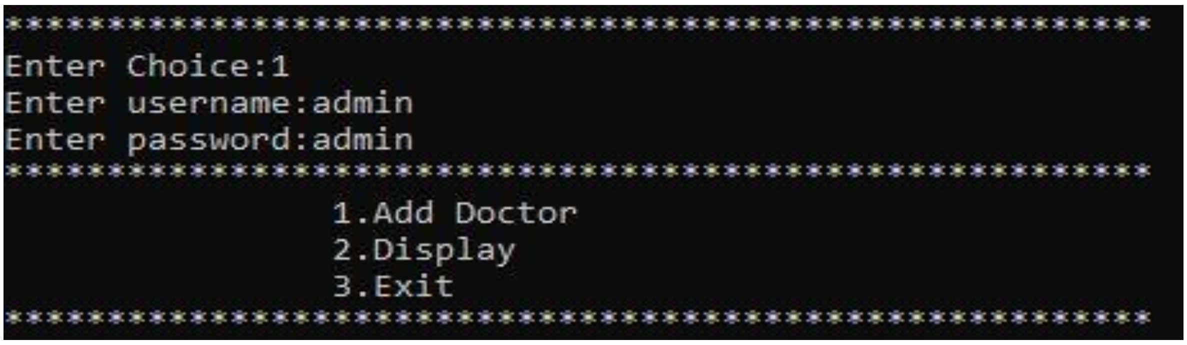
CHAPTER: 6

SNAPSHOTS



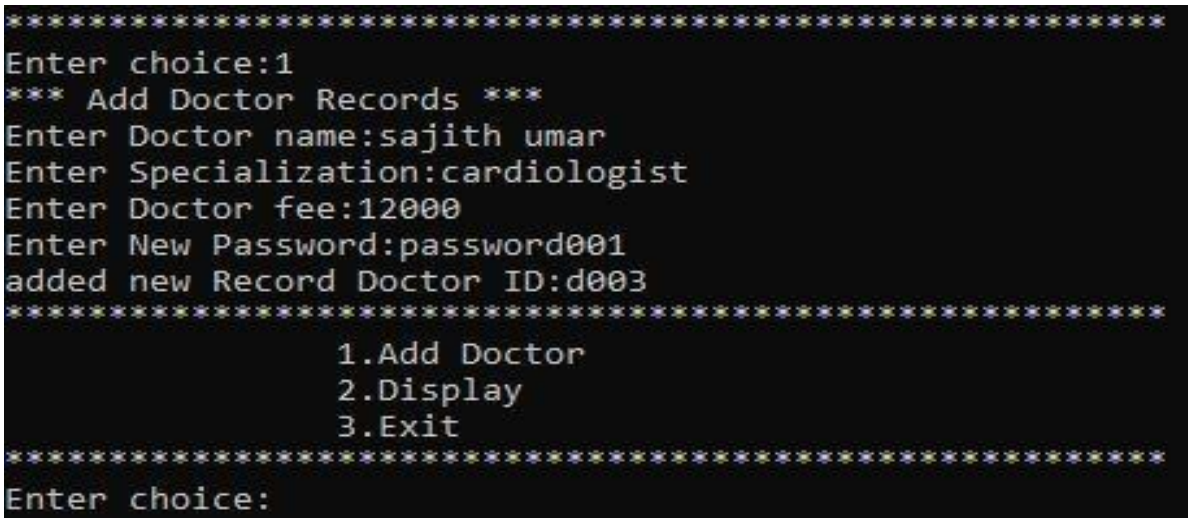
```
Command Prompt - python  hospital2.py
C:\Users\Umar>cd desktop

C:\Users\Umar\Desktop>cd fs mini project

C:\Users\Umar\Desktop\fs mini project>python hospital2.py
************** Hospital Management System **************
                1.Admin login
                2.Doctor login
                3.Patient login
                4.Exit
********************************************************
Enter Choice:
```

Fig 6.1: Home view



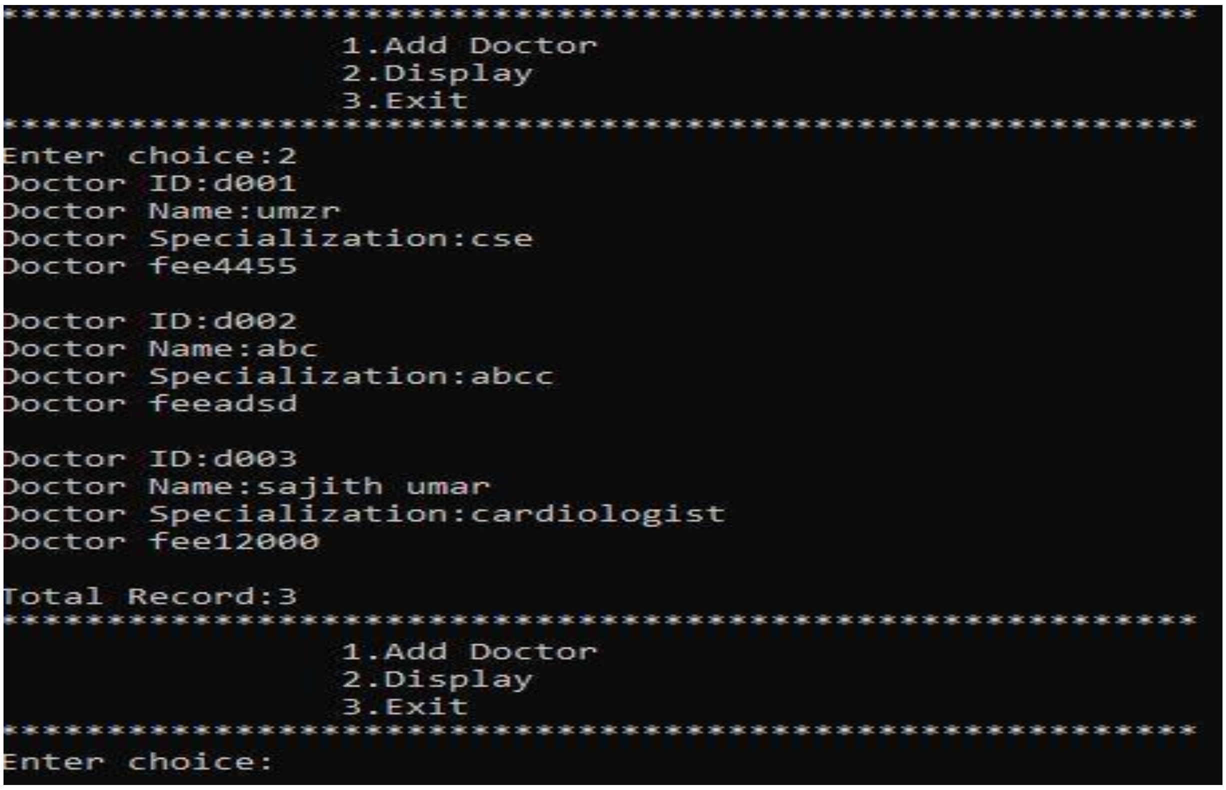
```
************************************************************
Enter Choice:1
Enter username:admin
Enter password:admin
************************************************************
                1.Add Doctor
                2.Display
                3.Exit
************************************************************
```

Fig 6.2: Admin login



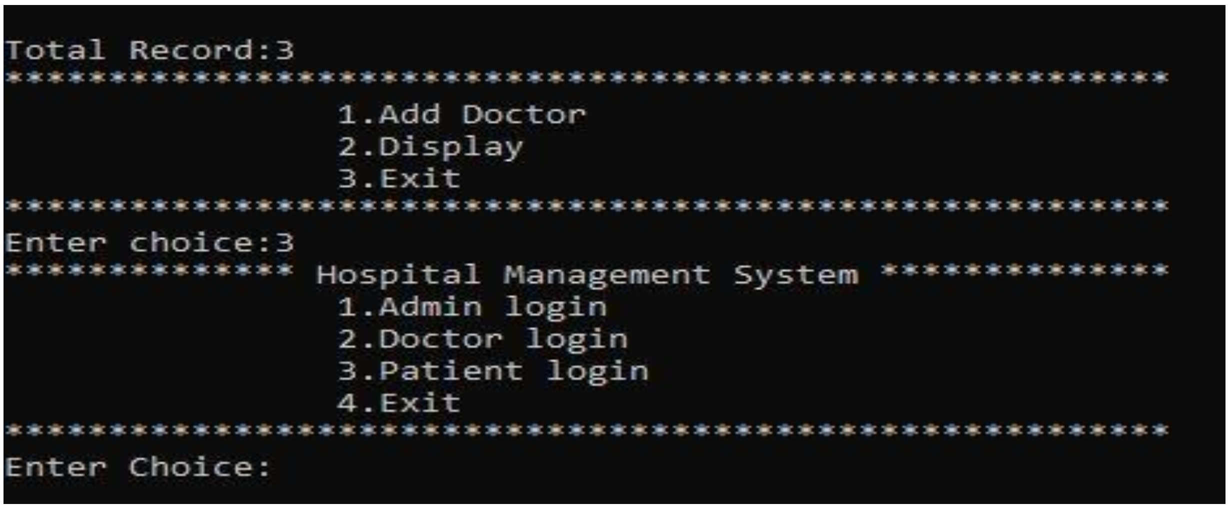
```
************************************************************
Enter choice:1
*** Add Doctor Records ***
Enter Doctor name:sajith umar
Enter Specialization:cardiologist
Enter Doctor fee:12000
Enter New Password:password001
added new Record Doctor ID:d003
************************************************************
                1.Add Doctor
                2.Display
                3.Exit
************************************************************
Enter choice:
```

Fig 6.3: Admin-add doctor



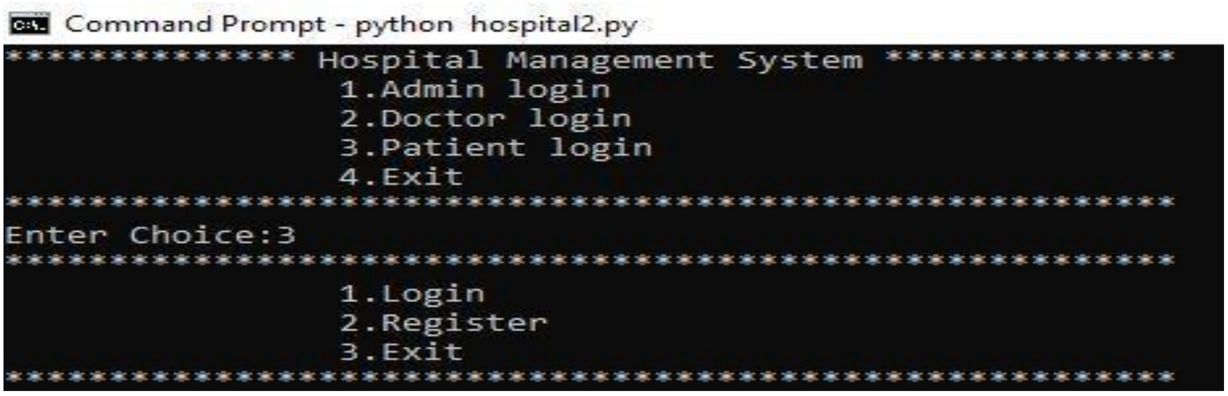
```
*********************************************************
                1.Add Doctor
                2.Display
                3.Exit
*********************************************************
Enter choice:2
Doctor ID:d001
Doctor Name:umzr
Doctor Specialization:cse
Doctor fee4455

Doctor ID:d002
Doctor Name:abc
Doctor Specialization:abcc
Doctor feeadsd

Doctor ID:d003
Doctor Name:sajith umar
Doctor Specialization:cardiologist
Doctor fee12000

Total Record:3
*********************************************************
                1.Add Doctor
                2.Display
                3.Exit
*********************************************************
Enter choice:
```

Fig 6.4: Admin-display doctor



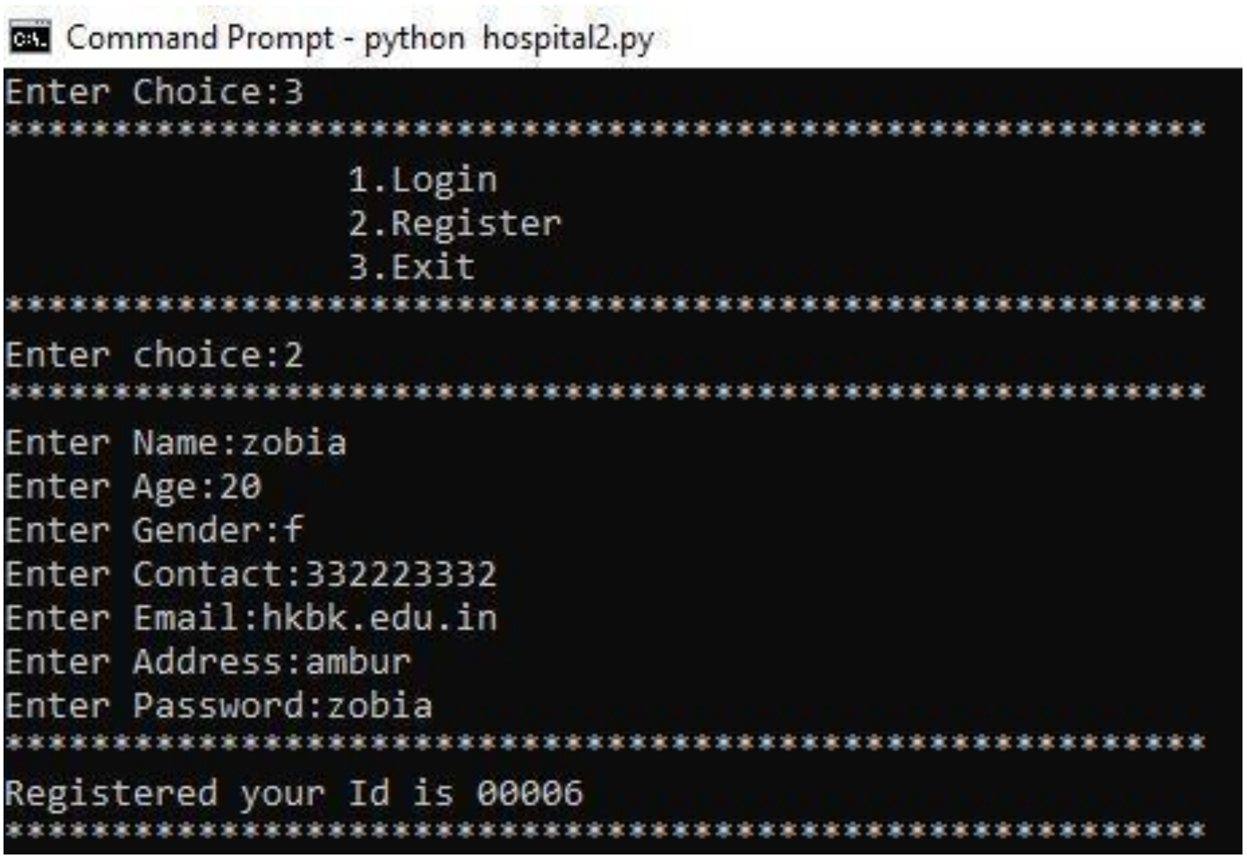
```
Total Record:3
*********************************************************
                1.Add Doctor
                2.Display
                3.Exit
*********************************************************
Enter choice:3
************** Hospital Management System **************
                1.Admin login
                2.Doctor login
                3.Patient login
                4.Exit
*********************************************************
Enter Choice:
```

Fig 6.5: Exit Admin



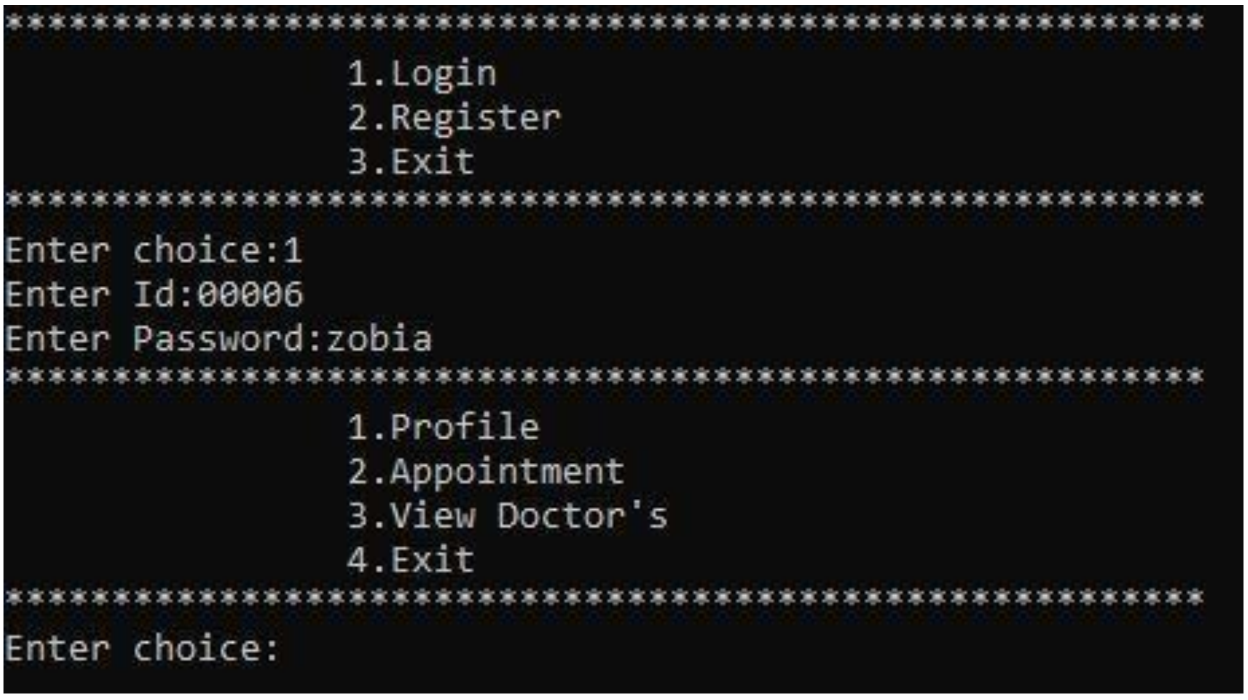
```
Command Prompt - python hospital2.py
************** Hospital Management System **************
                1.Admin login
                2.Doctor login
                3.Patient login
                4.Exit
*********************************************************
Enter Choice:3
*********************************************************
                1.Login
                2.Register
                3.Exit
*********************************************************
```

Fig 6.6: Patient login



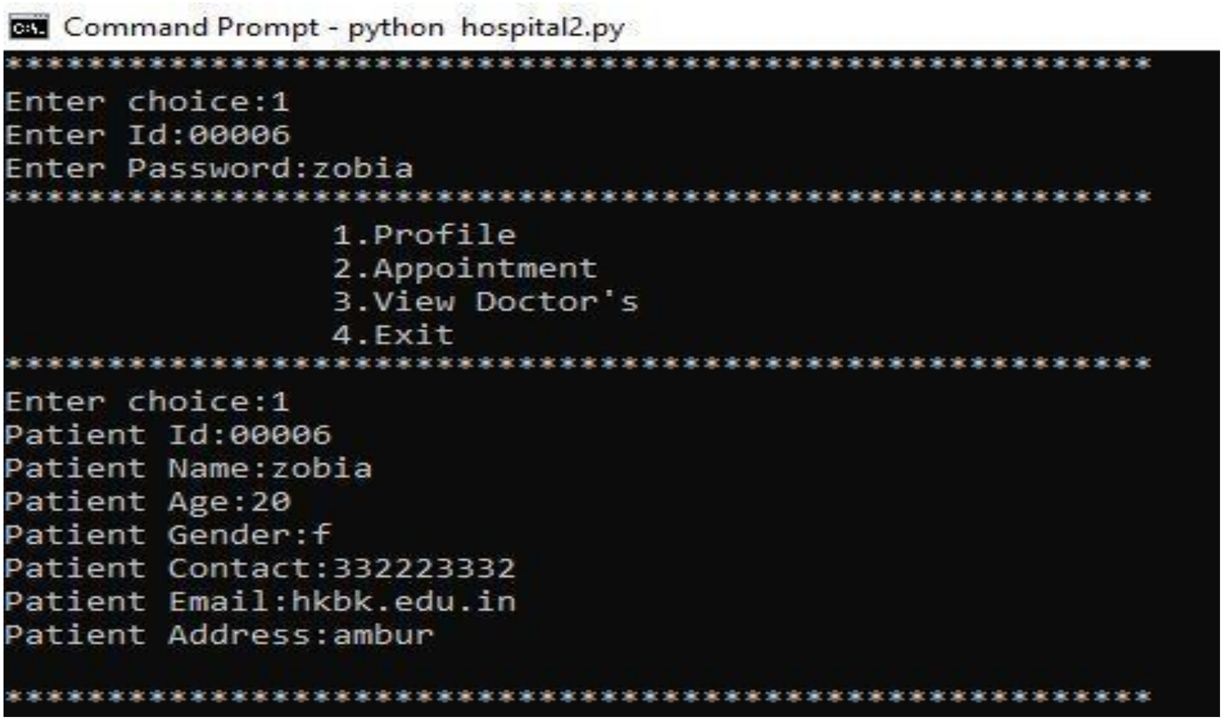
```
Command Prompt - python hospital2.py
Enter Choice:3
********************************************************
                1.Login
                2.Register
                3.Exit
********************************************************
Enter choice:2
********************************************************
Enter Name:zobia
Enter Age:20
Enter Gender:f
Enter Contact:332223332
Enter Email:hkbk.edu.in
Enter Address:ambur
Enter Password:zobia
********************************************************
Registered your Id is 00006
********************************************************
```

Fig 6.7: Patient register



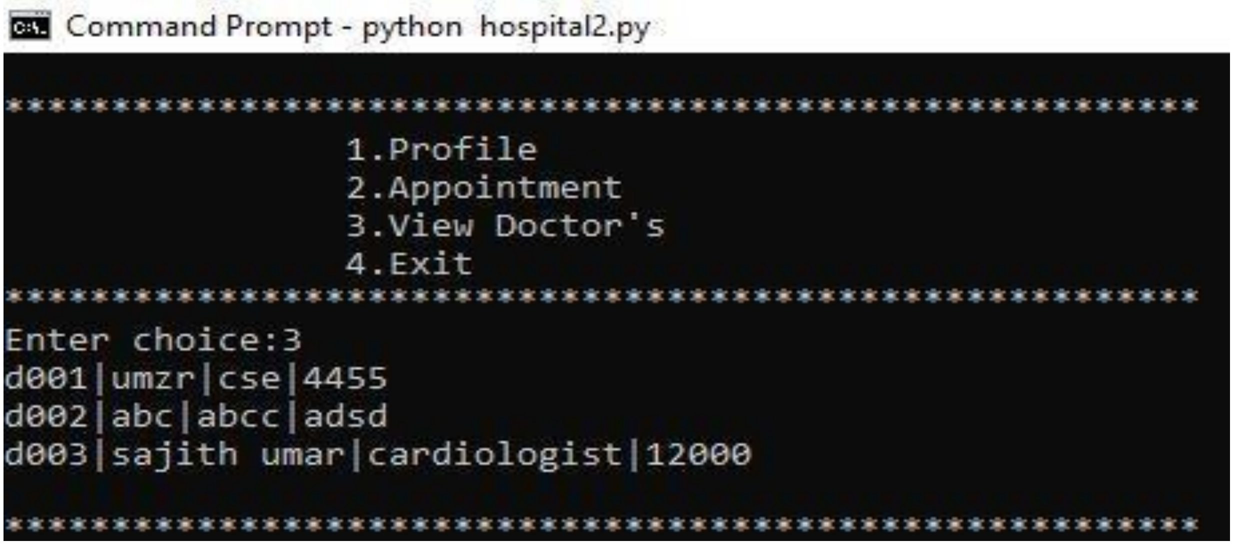
```
********************************************************
                1.Login
                2.Register
                3.Exit
********************************************************
Enter choice:1
Enter Id:00006
Enter Password:zobia
********************************************************
                1.Profile
                2.Appointment
                3.View Doctor's
                4.Exit
********************************************************
Enter choice:
```

Fig 6.8: Patient login-login



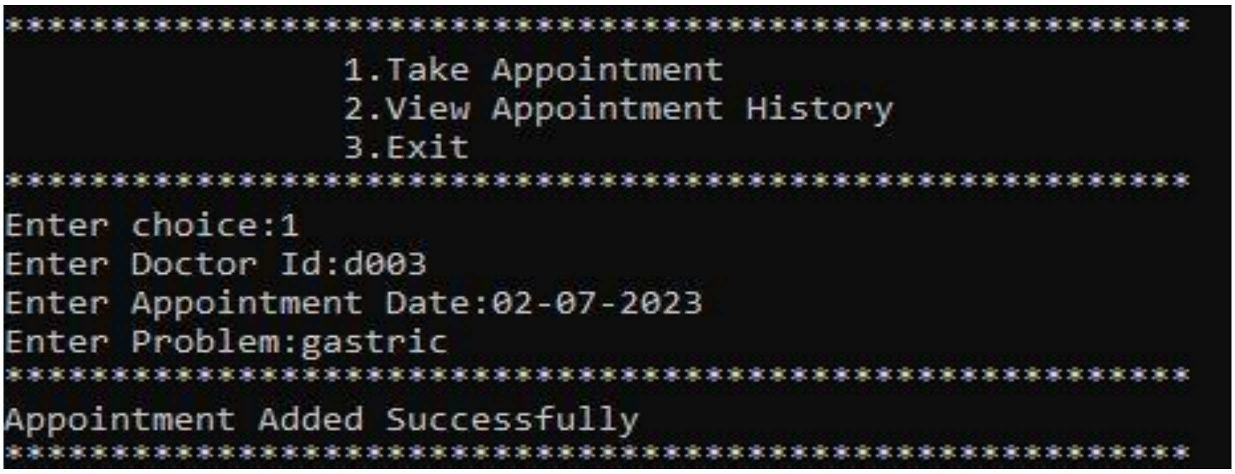
```
Command Prompt - python  hospital2.py
*********************************************************
Enter choice:1
Enter Id:00006
Enter Password:zobia
*********************************************************
                1.Profile
                2.Appointment
                3.View Doctor's
                4.Exit
*********************************************************
Enter choice:1
Patient Id:00006
Patient Name:zobia
Patient Age:20
Patient Gender:f
Patient Contact:332223332
Patient Email:hkbk.edu.in
Patient Address:ambur

*********************************************************
```

Fig 6.9: Patient profile



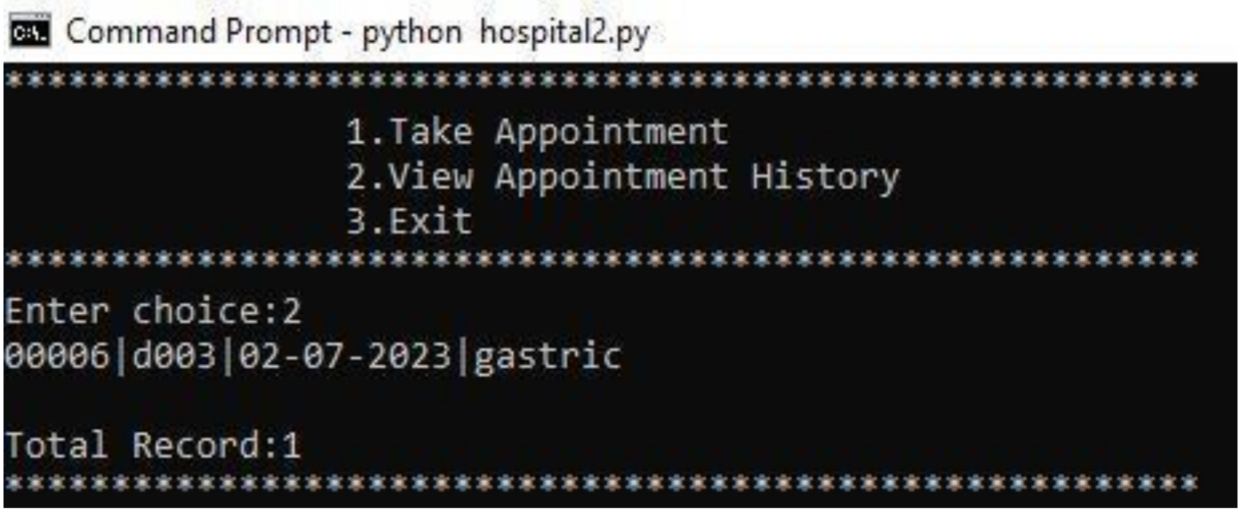
```
Command Prompt - python  hospital2.py

*********************************************************
                1.Profile
                2.Appointment
                3.View Doctor's
                4.Exit
*********************************************************
Enter choice:3
d001|umzr|cse|4455
d002|abc|abcc|adsd
d003|sajith umar|cardiologist|12000

*********************************************************
```

Fig 6.10: Patient -view doctors



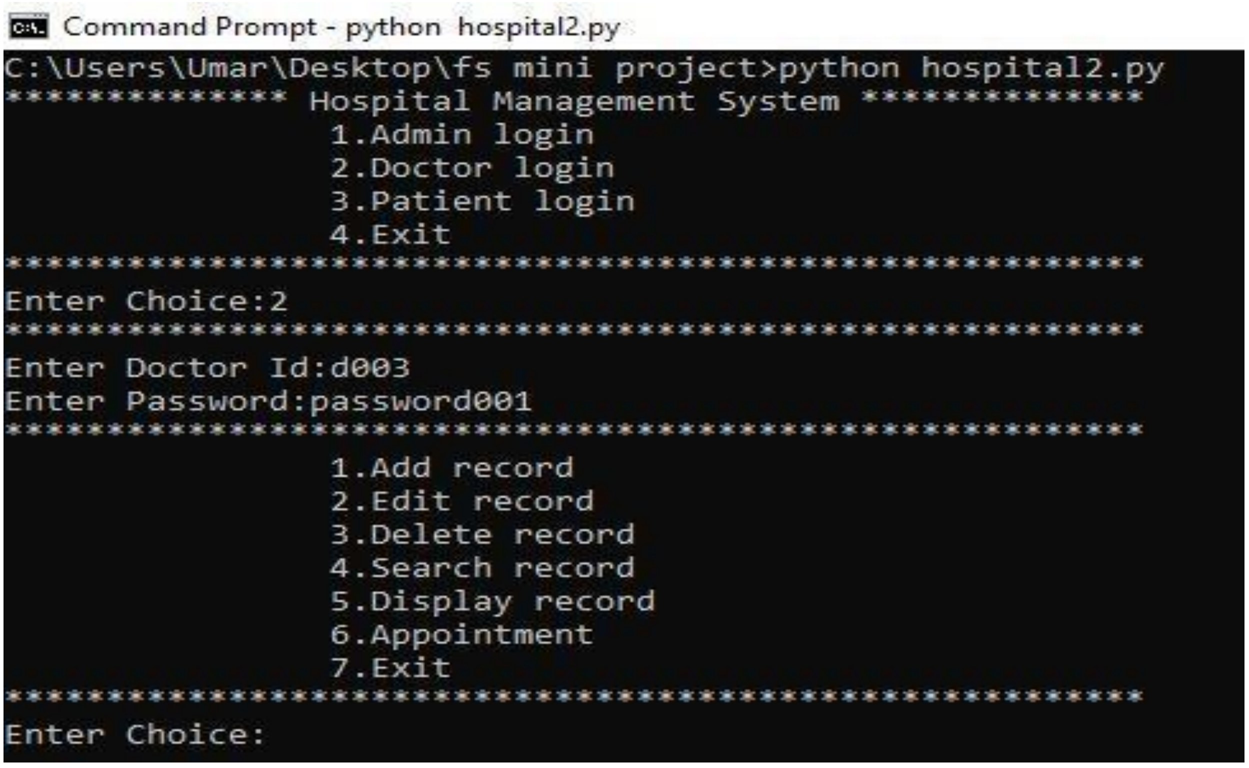
```
*************************************************************
                 1.Take Appointment
                 2.View Appointment History
                 3.Exit
*************************************************************
Enter choice:1
Enter Doctor Id:d003
Enter Appointment Date:02-07-2023
Enter Problem:gastric
*************************************************************
Appointment Added Successfully
*************************************************************
```

Fig 6.11: Patient booking appointment



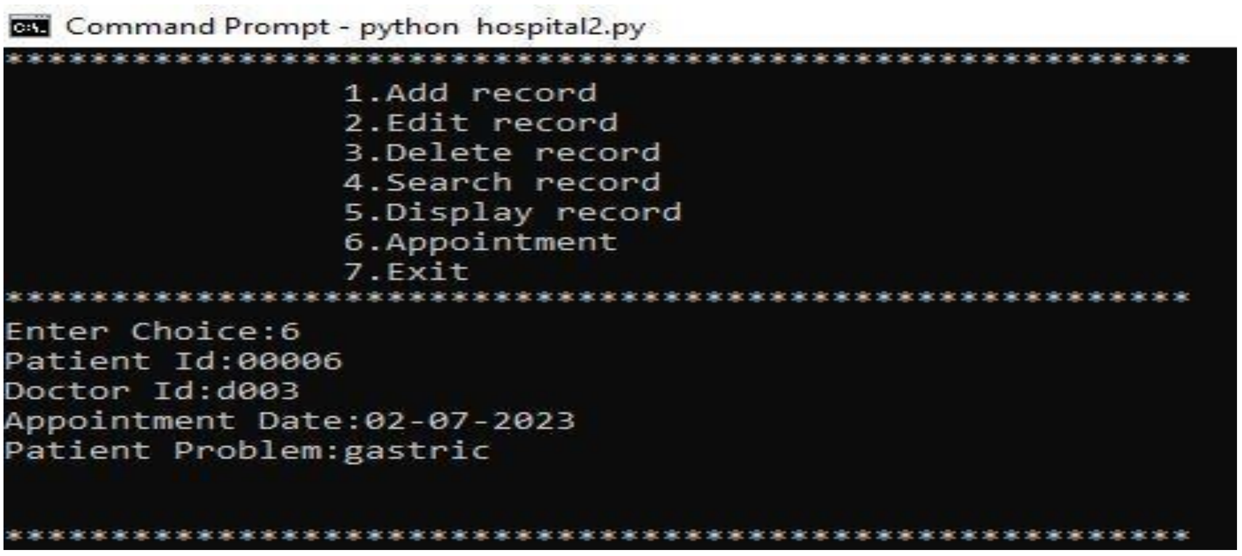
```
Command Prompt - python  hospital2.py
*************************************************************
                 1.Take Appointment
                 2.View Appointment History
                 3.Exit
*************************************************************
Enter choice:2
00006|d003|02-07-2023|gastric

Total Record:1
*************************************************************
```

Fig 6.12: Patient view appointment



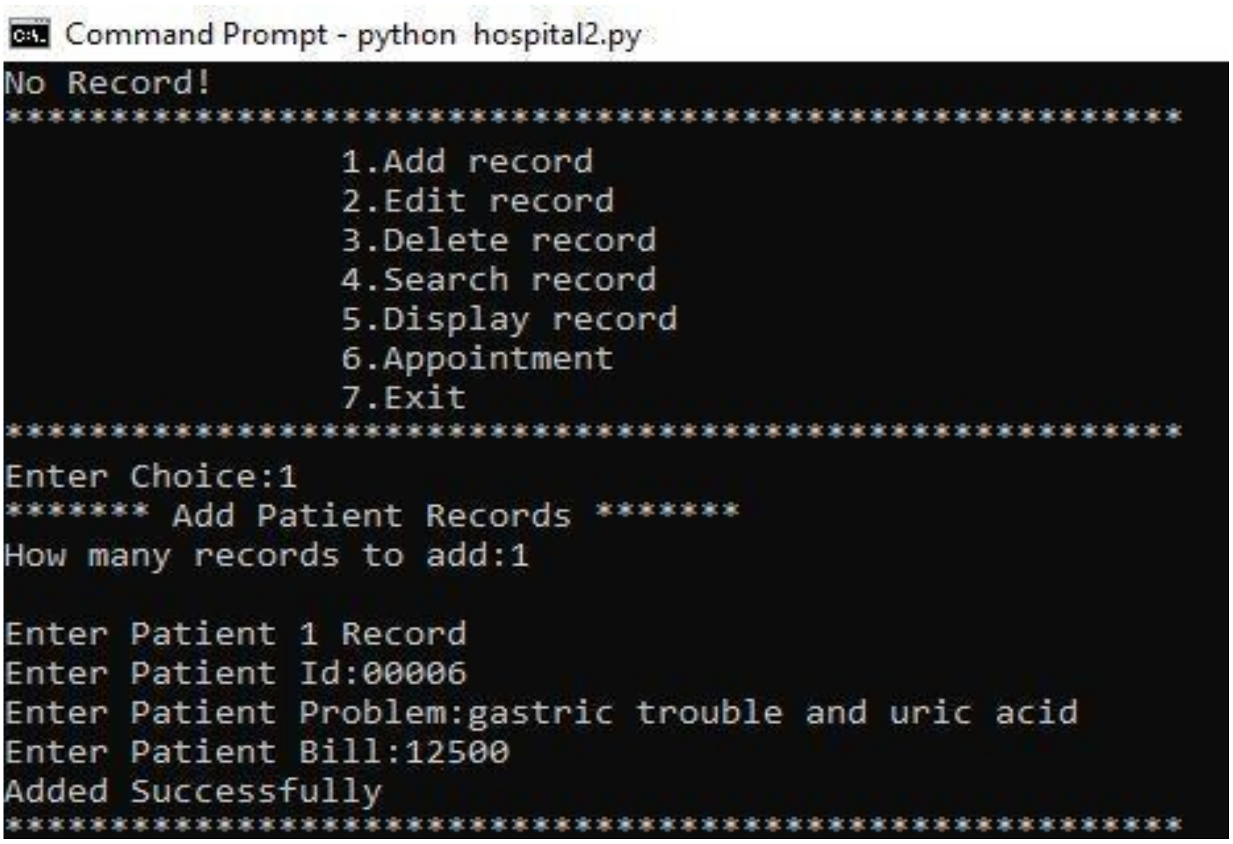
```
Command Prompt - python  hospital2.py
C:\Users\Umar\Desktop\fs mini project>python hospital2.py
*************** Hospital Management System ***************
                 1.Admin login
                 2.Doctor login
                 3.Patient login
                 4.Exit
**********************************************************
Enter Choice:2
**********************************************************
Enter Doctor Id:d003
Enter Password:password001
**********************************************************
                 1.Add record
                 2.Edit record
                 3.Delete record
                 4.Search record
                 5.Display record
                 6.Appointment
                 7.Exit
**********************************************************
Enter Choice:
```

Fig 6.13: Doctor login



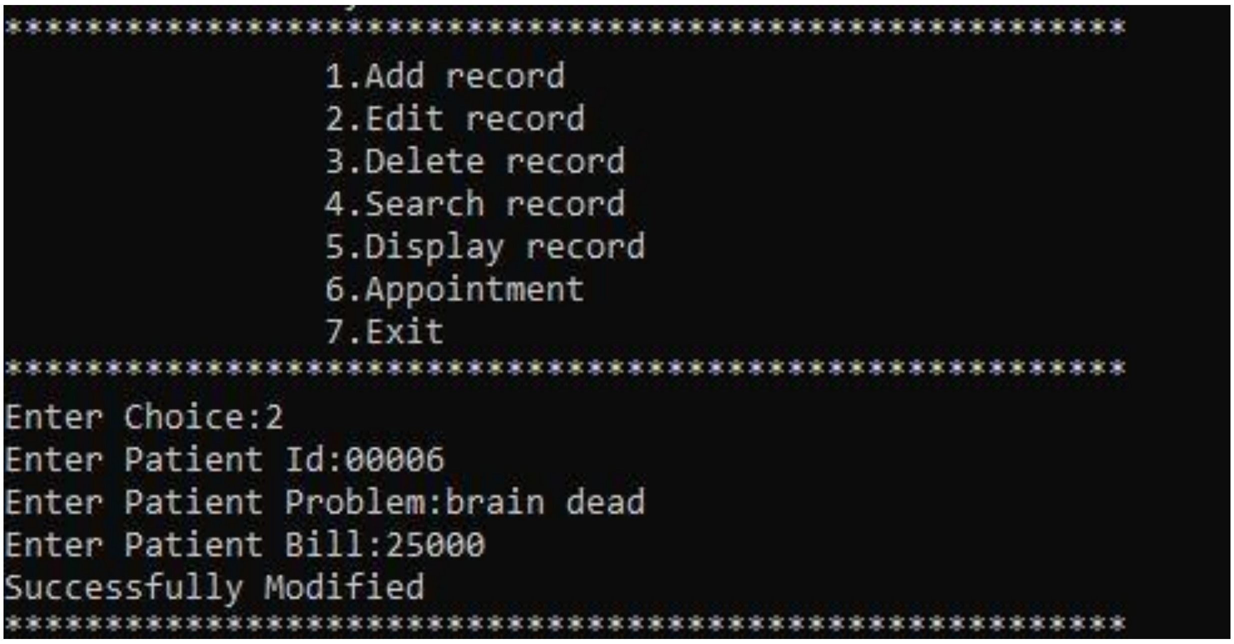
```
Command Prompt - python  hospital2.py
************************************************************
                     1.Add record
                     2.Edit record
                     3.Delete record
                     4.Search record
                     5.Display record
                     6.Appointment
                     7.Exit
************************************************************
Enter Choice:6
Patient Id:00006
Doctor Id:d003
Appointment Date:02-07-2023
Patient Problem:gastric


************************************************************
```

Fig 6.14: Doctor-view appointment booking



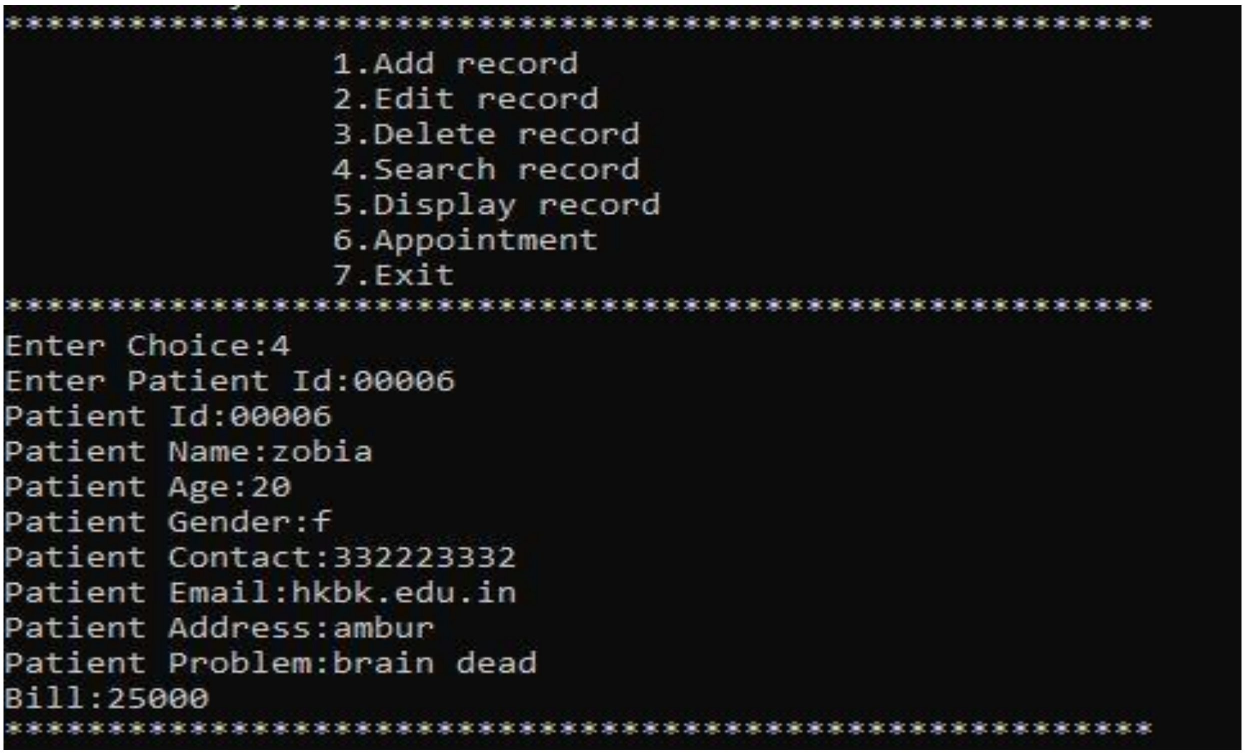
```
Command Prompt - python  hospital2.py
No Record!
*******************************************************
                1.Add record
                2.Edit record
                3.Delete record
                4.Search record
                5.Display record
                6.Appointment
                7.Exit
*******************************************************
Enter Choice:1
******* Add Patient Records *******
How many records to add:1

Enter Patient 1 Record
Enter Patient Id:00006
Enter Patient Problem:gastric trouble and uric acid
Enter Patient Bill:12500
Added Successfully
*******************************************************
```

Fig 6.15: Adding patient record



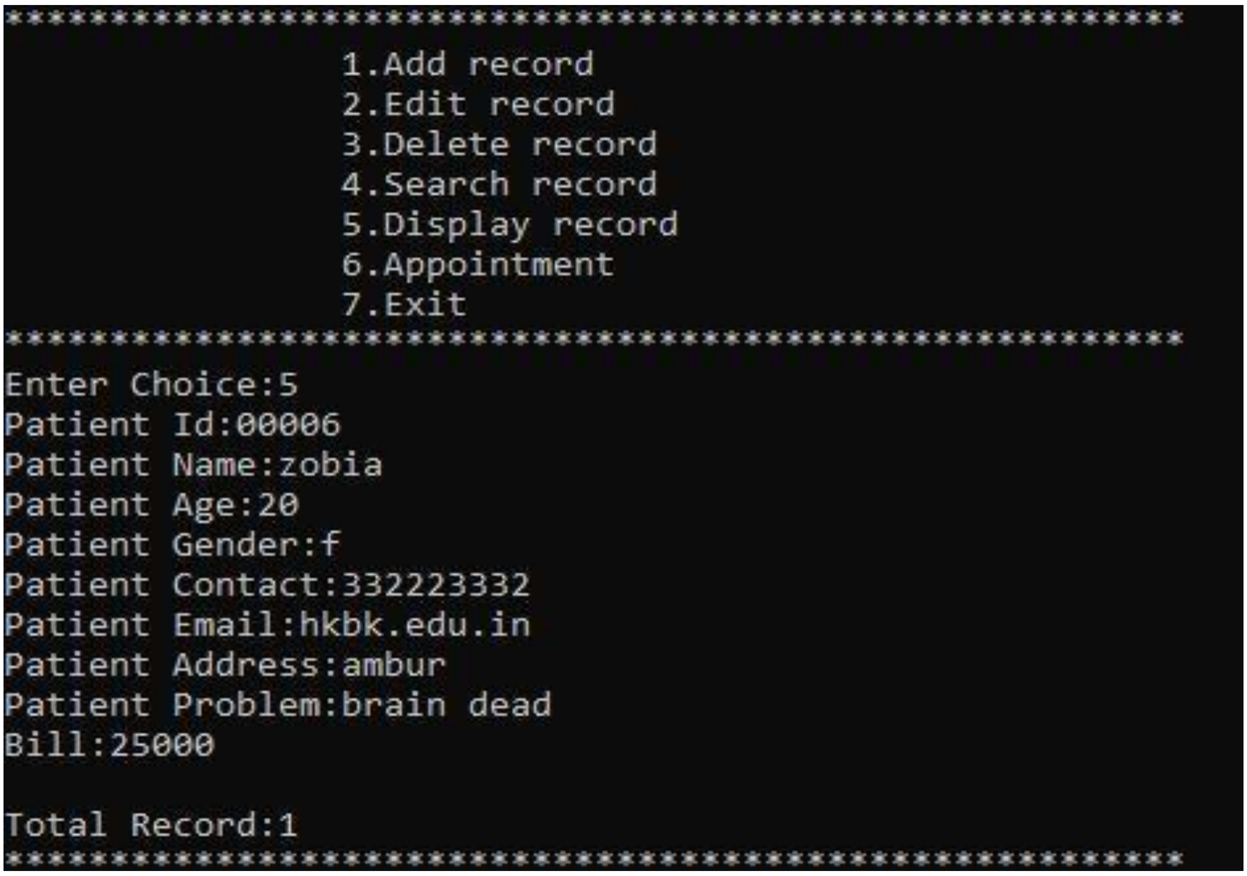
```
*******************************************************
                1.Add record
                2.Edit record
                3.Delete record
                4.Search record
                5.Display record
                6.Appointment
                7.Exit
*******************************************************
Enter Choice:2
Enter Patient Id:00006
Enter Patient Problem:brain dead
Enter Patient Bill:25000
Successfully Modified
*******************************************************
```

Fig 6.16: Edit patient record



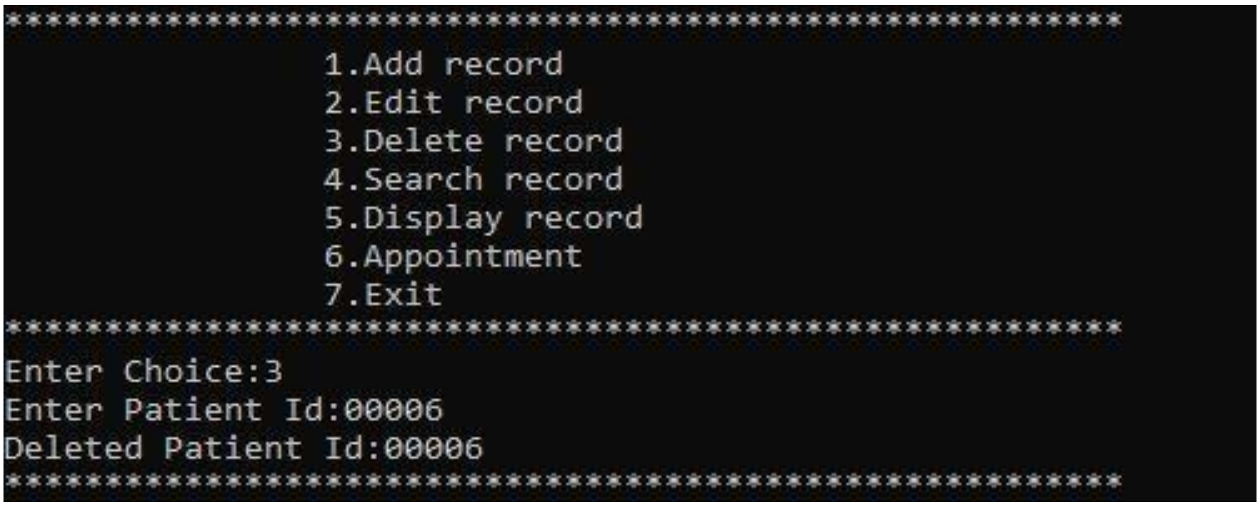
```
*********************************************************
                1.Add record
                2.Edit record
                3.Delete record
                4.Search record
                5.Display record
                6.Appointment
                7.Exit
*********************************************************
Enter Choice:4
Enter Patient Id:00006
Patient Id:00006
Patient Name:zobia
Patient Age:20
Patient Gender:f
Patient Contact:332223332
Patient Email:hkbk.edu.in
Patient Address:ambur
Patient Problem:brain dead
Bill:25000
*********************************************************
```

Fig 6.17: Searching patient record



```
*********************************************************
                1.Add record
                2.Edit record
                3.Delete record
                4.Search record
                5.Display record
                6.Appointment
                7.Exit
*********************************************************
Enter Choice:5
Patient Id:00006
Patient Name:zobia
Patient Age:20
Patient Gender:f
Patient Contact:332223332
Patient Email:hkbk.edu.in
Patient Address:ambur
Patient Problem:brain dead
Bill:25000

Total Record:1
*********************************************************
```

Fig 6.18: Displaying patient record



```
*********************************************************
                1.Add record
                2.Edit record
                3.Delete record
                4.Search record
                5.Display record
                6.Appointment
                7.Exit
*********************************************************
Enter Choice:3
Enter Patient Id:00006
Deleted Patient Id:00006
*********************************************************
```

Fig 6.19: Deleting Patient record

CONCLUSION

The “HOSPITAL RECORD SYSTEM” is successfully designed and developed to fulfill the necessaryrequirements, as identified in the requirements analysis phase, such as the system is very much user friendly, form level validation and field level validation are performing very efficiently. The present project has been developed to meet the aspirations indicated in the modern age. The Hospital RecordManagement System project has been successfully completed, providing an efficient and reliable solution for managing patient records within the hospital. Throughout the project, various objectives were achieved, leading to significant improvements in record-keeping processes and overall healthcare service delivery.

The project has been developed in a very short period of time and all efforts have been taken so that this project is very efficient in its execution there still exists some scope of improvement in our project.The following lists some of the enhancement that can be added incorporate into the project. It’s efficient software that includes all the basic functionalities like making data entries for new patients, and deleting and modifying records that are required for smooth functioning of a Hospital.

.

REFERENCES

1. "Design and Implementation of Hospital Management System" by S. Sujatha and M. Muthulakshmi (2015).

2. "Development of Hospital Records Management System with Barcode Technology" by M. B. Wibawa, I. U. Hidayanto, and A. N. Hidayanto (2017).

3. "Design and Development of a Hospital Records Management System" by F. M. Kakande and
   A. A. Ngugi (2018).

4. "Development of a Web-Based Hospital Information Management System" by B. C. Akamand J. U. Ekpoudom (2019).

5. "Evaluation of Hospital Records Management Systems: A Case Study" by S. M. Mathew and B.
   i. S. G. Sundarakumar (2020).

6. "Challenges and Opportunities for Hospital Information Systems: A Systematic Review" by H. Ammenwerth, C. Nitzlnader, and K. Hackl (2021).

7. "Development of Hospital Information Management System (HIMS): A Review" by N. Khan, M.
   A. Saad, and N. A. Shaikh (2020).

8. "Design and Implementation of Hospital Management System using Python" By, A. Adekoya, T.
   A. Omotayo, and O. E. Ayeni published in: International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5, Issue 8, August 2015.

9. "Development of an Electronic Medical Record System Using Django Framework and Python"By, M. A. Kumolu-Johnson, C. T. Akinwale, and M. O. Anwar published in: International Journal of Computer Science and Information Security, Vol. 14, No. 12, December 2016.

1.1 Introduction to file structure:

A file is a group of connected data that has a name and is stored on secondary storagesuch magnetic discs, magnetic tapes, and optical discs. Generally speaking, a file is a collection ofbits, bytes, lines, or records that have meanings determined by the user and creator of the file. The operating system's specified format should be followed when creating a file structure. Each typeof file has a specific defined structure. Text files are collections of characters arranged in lines.A source file is a list of steps and operations. An object file is a collection of data arranged into machine-understandable blocks.When an operating system specifies a different file structure, it also includes the supporting code. MS-DOS and Unix both provide a minimal number of file structures.It has to do with how you arrange your computer's files. It's a terrific approach to stay organised and make things simple for backing up your computer to keep all of your documents in one folder.An operating system also includes the supporting code when it defines an alternative file structure. Both MS-DOS and Unix offer a bare minimum of file structures.It has to do with how you organise the files on your computer. Keeping all of your documents in one folder is a great way to stay organised and make things simple while backing up your computer.

Various file structure formats are frequently employed in computer systems, including:

***Sequential File Structure*:**
In a sequential file structure, data is stored in a linear manner, with each record following the previous one. This structure is simple and easy to implement, whererecords are read or written sequentially from the beginning to the end. However, accessing or modifying specific records within the file can be time-consuming.

***Random File Structure*:**
Direct access to any record in a file is possible with a randomly generatedfile structure. The speedy retrieval and change of certain information is made possible by the unique identifier or key that is assigned to each record. Although random file formats can be moredifficult to build than sequential files, they are effective for applications that often retrieveindividual records.

***Indexed File Structure*:**

An index that maps the entries to their actual positions inside the file is part of an indexed file structure. The index normally consists of key-value pairs, where the key denotes a record's identifier and the value denotes its associated location. In contrast to sequential files, indexing enables quick retrieval of records based on their keys, facilitating speedier access.

***Hierarchical File Structure*:**

Data is arranged in a tree-like manner using a hierarchical file structure, with parent-child relationships between records. A hierarchy can be formed by the number of children each record can have. In file systems and databases, this structure is frequently used to describe the connections between data objects. Although they can be more difficult to handle, hierarchical file formats provide for quick exploration and retrieval of related items.

***Network File Structure*:**

The network file format provides relationships between records similarly to a hierarchical structure, but it permits more flexible connections. Records in this format can have numerous parent and child records, resulting in a structure that resembles a network. When working with network databases, where data items may have various relationships, this kind of file structure is frequently employed.These are but a few illustrations of file structures; in accordance with particular requirements, more modifications and combinations are also possible. The type of data, access patterns, storage efficiency, and system performance requirements all have an impact on the choice of file structure. For effective data storage, retrieval, and management within computer systems, understanding file formats is crucial.

1.2 Introduction to the project:

Our project, "Hospital Record System Using Data Processing and Indexing," aims to computerize hospital front office management and provide software that is user-friendly, straightforward, quick, and economical. It deals with gathering data about patients, doctors, receptionists, and other parties, such as diagnosis specifics. The current manual technique is cumbersome, slow, and prone to mistakes; it should be computerized for more rapid, effective results and satisfied customers. Traditionally, it was carried out by hand. The system's primary purpose is to register, store, and retrieve patient and physician information as needed, as well as to change this information in a meaningful way.

The project's goal is to use data processing and indexing techniques to create a reliable and effective hospital record management system. The maintenance of patient records is crucial in the modern healthcare sector for guaranteeing high-quality service and effective operations. Adoptingcutting-edge technologies is now necessary to simplify the record management process due to the growing amount of patient data being generated everyday.

The suggested system will make efficient use of data processing methods to manage enormous volumes of patient data. To efficiently store, retrieve, and process patient records, it will make use of advanced algorithms and data structures. The system will be able to handle sophisticated queries and produce insightful data analysis by utilizing data processing techniques. As a result, healthcare workers will be able to swiftly access pertinent patient information and make well-informed decisions, thereby improving patient care.

The system will also include indexing mechanisms to improve the accuracy and speed of data retrieval. Making data structures for indexing entails enabling effective search operations based on numerous attributes, such as patient names, medical conditions, or admission dates. The system can greatly cut down the time needed to get specific information by indexing the records, increasing the overall effectiveness of the hospital's operations. Additionally, indexing will make it simple to spot redundant or duplicate records, removing any potential data discrepancies and guaranteeing data integrity.

Data security and confidentiality will also be given top priority in the project. Protecting the data found in patient records, which includes private and sensitive information, is essential. To ensure that only authorised personnel may access and edit the records, the system will implement strong security features, such as user authentication, access limits, and encryption.

A primary focus in the development process will be adherence to pertinent data privacy laws and industry best practices. Therefore, the hospital record management system will transform how patient records are kept by utilizing data processing and indexing techniques. The system will enhance the effectiveness, accuracy, and security of record management procedures by utilizing cutting-edge algorithms, data structures, and indexing techniques. As a result, patient care will be improved, informed decision- making will be possible, and hospital healthcare services will function more effectively as a whole.

1.3 Objectives of the project

Efficient Data Processing: By automating numerous operations like data entry, storage, retrieval, and updating, the system attempts to streamline the processing of medical information. This goal is to increase efficiency while minimizing human labor and error-proneprocesses.

Centralized Record Management:

The system intends to centralize medical record storage andmanagement, making sure that all pertinent data is quickly available from a single platform.This goal contributes to greater organization and accessibility by removing the need fornumerous physical or electronic file systems.

Enhanced Data Indexing:

The system puts a lot of emphasis on using efficient indexing methods to organize and classify healthcare records. This goal permits the rapid and precise retrieval of particular records based on a variety of criteria, including patient information, a diagnosis, a course of treatment, dates, and more. It speeds up effective search processes anddecreases the time needed to find pertinent records.

Data Security and Privacy:

The system uses the proper access controls, encryption safeguards, and user authentication processes to protect the confidentiality and privacy of patient records. By maintaining compliance with data protection rules like HIPAA (HealthInsurance Portability and Accountability Act) or other relevant laws, this goal helps to safeguard private patient data from unauthorized access.

Integration with Existing Systems:

The system seeks integration with other hospital systems,including electronic health record (EHR) systems, laboratory information systems, billing systems, and others. This goal promotes smooth data exchange and system interoperability, which lessens duplication and boosts overall operational effectiveness.

Reporting and Analytics:

The system promises to offer thorough reporting and analytics capabilities, enabling hospital workers and management to learn from the data. Improvedpatient care and operational results are the result of this goal's facilitation of data-driven decision-making, trend analysis, resource planning, and performance evaluation.

Scalability and Future Expansion: In order to accommodate expansion and technological improvements in the future, the system aspires to be scalable and adaptive. This goal makessure the system can accommodate more capabilities or modules as needed while handling growing data volumes.

1.4 Scope of the project

The project involves designing and deploying a holistic and effective digital solution for managing hospital records. This system is designed to handle the storage, arrangement, retrieval, and administration of patient records within a medical facility. Its primary goal is to enhance operational efficiency by automating administrative and clinical procedures related to patient data. The system ensures precise data accuracy, confidentiality, accessibility, and optimized workflow. Additionally, the project entails developing a user-friendly interface that allows authorized staff members to securely input, update, and access patient information.

The project encompasses the integration of diverse modules, including electronic health records (EHR), appointment scheduling, billing and invoicing, laboratory and diagnostic results, prescription management, and medical imaging, among others. These modules are brought together into a unified system. Moreover, the scope includes the implementation of robust data security measures to safeguard patient privacy and comply with regulatory standards such as the Health Insurance Portability and Accountability Act (HIPAA). The project also involves staff training on system usage and the provision of ongoing technical support to ensure smooth operation and maintenance. The ultimate objective of the hospital record management system is to improve the overall quality of patient care, facilitate effective decision-making for healthcare professionals, reduce errors and redundancies, and enable seamless information exchange amongdifferent hospital departments.

1.5 History and evolution of technology used

1. Python:

- History: Python is a versatile and widely used programming language that was created by Guido van Rossum and first released in 1991. Its development was motivated by the desire for a language that emphasized simplicity and readability, making it accessible to both novice and experienced programmers. Python gainedpopularity due to its clean and concise syntax, extensive standard library, and strong community support. Over the years, it has evolved and adapted to meet thechanging needs of developers, incorporating new features and improvements. Python's flexibility and ease of use have made it a preferred language for a varietyof applications, ranging from web development and scientific computing to artificial intelligence and data analysis. Today, Python is one of the most widely used programming languages, with a thriving ecosystem and a vibrant communityof developers contributing to its growth and success.

- Evolution: Python has undergone significant evolution since its inception. Initially released in 1991, Python started as a simple and intuitive programming language with a focus on readability and ease of use. Over the years, it has experienced several major releases that introduced new features and improvements. Python 2, released in 2000, brought many enhancements and became widely adopted. However, as Python 2 reached its end-of-life in 2020, Python 3 emerged as the successor, addressing various design flaws and introducing backward-incompatible changes. Python 3 introduced improvements like better Unicode support, enhanced syntax, and performance optimizations. It also fostered a stronger community around the language, leading to the development of numerous libraries, frameworks, and tools. Python's versatility and extensive library ecosystem have contributed to its popularity in various domains, including web development, data analysis, machine learning, and scientific computing. In recent years, Python has continued to evolve, with regular updates introducing new features, enhancements, and performance optimizations, making it one of the most widely used and versatileprogramming languages today.

2.Command Prompt:

- History: The command prompt, also known as the command line interface (CLI), has a rich history in the world of computing. It can be traced back to the early days of computer operating systems, where text-based interfaces were the primary meansof interacting with a computer. One of the earliest command prompt implementations was the COMMAND.COM shell in MS-DOS, introduced by Microsoft in the early 1980s. This allowed users to execute commands by typing them directly into the command prompt. Similar command prompt interfaces were present in other operating systems like Unix.

- Evolution: In the early days of computing, command prompts were primarily usedby computer professionals and developers. They were often limited to a single lineof text, requiring users to type commands precisely. These prompts relied on specific commands and syntax, with minimal interactive features. As computer technology advanced, command prompts became more sophisticated. The introduction of graphical user interfaces (GUIs) in the 1980s brought a shift towards mouse-driven interactions and visual representations. However, commandprompts continued to play a vital role for power users and system administrators who required direct control and automation capabilities. the evolution of the command prompt has transformed it from a simple text

-based interface to a powerful tool with advanced features and capabilities. While graphical interfaces have become the norm for most users, the command prompt remains a valuable toolfor professionals and power users, enabling efficient system management, automation, and scripting.

3. Data Structures:

- History: For many years, computer science has investigated and used data structures. With contributions from several scholars and practitioners, numerousdata structures have been created and improved over time.
- Evolution: Tuples, Queues,Hash tables in Python and Array List are just a few of the data structures offered by Python's standard library. These data structures have been developed to provide better performance, better memory use, and more features. Additionally, third-party frameworks and libraries like Google Guava andApache Commons Collections offer specialized data structures and tools.

Performance Analysis:

- History: Software development has long been concerned with performance analysis. The performance of apps has been measured and optimized using a varietyof ways.
- Evolution: Performance analysis approaches have changed along with improvements in hardware and software. Execution time, memory use, and other performance indicators have been measured using profiling tools, benchmarking frameworks, and performance monitoring tools. These technologies have improved in sophistication and now offer thorough analysis and insights for optimization.

1.6 File Indexing History

- For a long time, file indexing has been a crucial part of information retrieval systems. The most of file indexing methods used in the early days of computing used straightforward linear procedures, which involved sequentially scanning files to create an index. However, more complex indexing methods were created to improve efficiency and search capabilities as technology evolved and the volume ofdata grew.
- with the development of data structures and algorithms, file indexing significantly improved. Tree indexing, which became a potent method for effectively storing andsearching words, was one such advancement. aims to reduce the search space and enable quick prefix-based searches by using a tree-like structure where each node represents a word's letter. This method has proven to be especially helpful in programs like word editors, spell checkers, and search engines.
- File indexing algorithms are still developing and adapting to the ever-growing volume and complexity of data in the current technological environment. Advanced indexing methods like inverted indexing and compressed indexing have become more popular as a result of the growth of massive data and the requirement for real-time information retrieval.
- Search engines frequently employ inverted indexing to effectively manage big document collections. It creates an index that associates words or keywords withthe content of the documents, enabling quick retrieval of pertinent documents in response to search queries. Ranking algorithms are frequently used in inverted indexing approaches to order the retrieved documents according to relevance.
- Compressed indexing methods concentrate on lowering the index's storage needs while keeping effective search capabilities. The index structure is compressed using a variety of compression algorithms, including delta encoding, front coding, and variable-length encoding. When dealing with big data sets or situations where storage capacity is at a premium, these strategies are very useful.
- Finally, file indexing methods have advanced significantly from the early days of linear indexing. The field has made great strides, from the advent of tree indexing and B+ tree indexing to the developments of inverted indexing and compressed indexing. File indexing will continue to develop in response to the problems of the digital era as a result of the convergence of hardware improvements and breakthroughs in algorithms and data structures. File indexing continues to be a keycomponent of effective information retrieval.

7

LITERATURE SURVEY

In a project report, a literature survey or literature review section presents an analysis of existing research and publications related to the field of interest. It examines the findings, methodologies, and outcomes of previous studies while considering the project's specific parameters and scope. This section provides a comprehensive overview of the current knowledge and understanding of the subject matter, highlighting the gaps, limitations, and potential areas for further exploration within 3 the project. It is the most important part of your report as it gives you a direction in the area of your research.It helps you set a goal for your analysis - thus giving you your problem statement.

2.1 Existing system

To save the information on patients and doctors, the hospital administration system currently keepsmanual records in the form of registers, folders, etc. Making a new appointment requires a lot of paperwork. 4 Additionally, it causes data redundancy. The analysis phase comprises a thorough examination of the current system.

Limitations:

- Time delay Redundancy, Accuracy
- Information Retrieval.
- Storage Media

All these are rectified in the proposed system. Much emphasis is given upon the process of patient servicing.

Drawbacks associated with existing systems:

***Paper-based Documentation*:**

Many hospitals still keep patient records on paper-baseddocumentation. This can cause a number of problems, including lost or misplaced documents, trouble finding and retrieving information, and increasing storage needs.

***Limited Accessibility*:** Accessing patient records in a system that relies on paper can be laborious and time-consuming. Delays in patient treatment and decision-making may result from healthcare providers having to physically find and retrieve files.

***Data Security Risks*:** Paper documents are susceptible to loss, theft, or damage. Patient privacy and confidentiality may be jeopardized if patient information is managed improperly, lost, oreven stolen.

***Inefficient Workflow*:** Manually processing paper documents can bog down administrative procedures and make medical workflows inefficient. Staff workers may spend more time caring for patients instead of spending so much time on activities like filing, retrieving, andorganizing records.

***Lack of Integration*:** Electronic health record (EHR) systems and other hospital systems are frequently not integrated into paper-based systems. This may make it more difficult for healthcare providers to work together, share data for research, and use analytics.

***Limited Data Analysis Capabilities*:** It is difficult to carry out in-depth data analysis and derive valuable insights from paper records. The manual nature of data processing makes it time- consuming and less accurate to analyze trends, spot patterns, and carry out quality improvementprogrammes.

***Higher Costs*:** Paper, printing, storage, and other physical infrastructure costs are continuing costs associated with maintaining a paper-based record management system. Additionally, thelabour expenses associated with manual operations may rise as a function of how time- consuming they are.

***Difficulty in Disaster Recovery*:** Paper documents are easily lost or destroyed in emergencysituations, such as a fire or natural disaster. Critical patient information becomes extremelydifficult to recover and restore, potentially putting patient care at risk.

***Incomplete or Inaccurate Data*:** Paper records are susceptible to human error in data entry,illegible handwriting, or missing information. These mistakes may result in erroneous or incomplete patient data, which could have an impact on clinical judgement and patient safety.

2.2Proposed system

The current method uses more manpower than is necessary and has apparent flaws. The currently suggested system is fairly simple to use. The amount of work done manually will decrease when each department in the health center is computerized. The minimum amount of manpower is used.Because there is never a need to go through the files on the shelves, the patients at the registration office are registered quickly. In addition to avoiding redundant and inconsistent data, it is quick, effective, and dependable.

The adoption of an electronic health record (EHR) system is one of the main components of the suggested system. The necessity for paper-based records will be eradicated since healthcare providers will be able to keep and retrieve patient information digitally. A centralized and thoroughrepository of patient information, including medical history, test results, diagnoses, prescriptions, and treatment plans, will be made available by the EHR system. By enabling rapid and simple access to patient information, qualified healthcare providers will be able to make decisions more quickly and accurately.

The suggested system will include strong security features to guarantee data protection and privacy. Role-basedauthentication and encryption mechanisms will be used to tightly regulate access to patient records. In order to protect patient anonymity, the system would also adhere to industry standards and laws including the Health Insurance Portability and Accountability Act (HIPAA). Interoperability will be given top priority in the planned system, facilitating easy data sharing between varioushealthcare systems and providers. Improved patient care continuity and improved professional teamwork will result from this, especially when patients receive treatment from various healthcare providers.

Healthcare workers will find it simple to retrieve and update patient records because to the system's user-friendly interfaces and intuitive navigation. Additionally, it will feature sophisticated search features that let users easily find particular data in the sizable record database. Additionally, the proposed system would use data analytics and reporting tools to derive useful insights from thegathered data, which can assist healthcare organisations in identifying trends, tracking patient outcomes, and making defensible decisions to improve healthcare delivery.

Therefore, the proposed hospital record management system aims to enhance efficiency, data security, and patient care within healthcare institutions. By leveraging advanced technology and streamlined processes, the system will provide healthcare professionals with easy access to comprehensive patient records, promoting better decision-making and improved healthcare outcomes.

2.3Advantages of proposed system

***Improved Efficiency*:** The suggested solution streamlines all aspects of record management,making it possible to handle patient data more quickly and effectively. This results in less paperwork, quicker record retrieval, and enhanced efficiency throughout the institution.

***Enhanced Data Security*:** To safeguard patient records from unauthorized access or breaches, the system offers strong data security measures, such as encryption and access limits. This ensures the confidentiality of sensitive information and lowers the possibility of data loss.

***Accurate and Complete Documentation*:** By implementing the proposed system, hospitals can ensure accurate and comprehensive documentation of patient records. The system typically includes standardized templates and prompts, reducing the likelihood of missing or incomplete information.

***Easy Accessibility and Retrieval*:** The ability to quickly access and retrieve patient data is made possible by digitizing records. Authorized healthcare practitioners may rapidly look up specific records, saving time on laborious searches and facilitating swift decisions.

***Integration and Interoperability*:** Electronic health records (EHRs) and other hospital systems and equipment, as well as medical devices, are frequently able to be integrated with the proposed system. Through enabling seamless data transmission and enhancing overall healthcare coordination, this integration fosters interoperability.

***Data Analytics and Reporting*:** Through data analytics, the system can produce useful insights with the digitization of documents. Hospitals are able to do trend analysis, spot patterns, and produce reports that aid in research, quality improvement projects, and well-informed decision-making.

***Cost and Space Savings*:** By switching to a digital system instead of paper-based records, physical storage space is not required, and printing and paper costs are also decreased. The suggested solution reduces the possibility of losing or misplacing records, potentially saving the hospital money.

***Improved Communication and Collaboration*:** The system often has tools that let medical professionals interact and work together efficiently. It can enable real-time updates, secure document sharing via messaging, and effective teamwork amongst employees.

***Patient Empowerment*:** Patients can view their medical information, test results, and treatment plans through patient portals or access to electronic health records, which are frequently included in the proposed system. As a result, there is increased patient involvement, patient empowerment, and improved patient-provider communication.

***Regulatory Compliance*:** The system can help hospitals meet regulatory requirements and standards related to record keeping, privacy, and data security. It assists in adhering to guidelines such as the Health Insurance Portability and Accountability Act (HIPAA) and ensures compliance with legal obligations.

It's important to note that the advantages mentioned above may vary depending on the specific featuresand capabilities of the proposed hospital record management system.

3.1 HARDWARE REQUIREMENTS

1. CPU: For best performance, a recent multi-core CPU such the Intel Core i5 or anequivalent is advised. Indexing and analysis operations for the project can take advantage of the processing power provided by multi-core platforms.

2. Memory (RAM): To ensure efficient operation, the project needs at least 4 GB of RAM. Higher RAM capacity, such as 8 GB or more, may be advantageous for better performance, depending on the amount of the files being indexed and the complexity ofthe indexing techniques.

3. Storage: The project files, including the input files, index files, and any generated reports or logs, must be kept in a storage location with enough room. The size and quantity of the files being indexed determine the actual storage needs. It is advised tohave several gigabytes of unoccupied storage space.

4. Display: To view the user interface and outcomes of the project, a monitor or display screen with a resolution of at least 1024x768 pixels is required. The user experience may be improved by higher resolutions, especially when working withhuge amounts of data.

5. Input Devices: For engaging with the project's user interface and entering data,such as file selections and setup options, a normal keyboard and mouse are required.

6. Network Connectivity: Internet access is not necessary for the project's fundamental operation in terms of network connectivity. However, a reliable internet connection is required if the project calls for collecting files from distantlocations or using online resources for comparison.

3.2 SOFTWARE REQUIREMENTS

1. Python : A compatible Python version must be installed because the project was developed usingthe Python programming language. Here, we have used Python version 3.11.0.

2. Integrated Development Environment (IDE): For effective project development and debugging, an IDE like Visual studio code, Pycharm, or Pydev or Commandprompt is advised. Here, we have Command prompt also known as cmd.exe or cmd.

3. Libraries and Dependencies: Depending on the indexing methods selected and any implemented specific functionalities, the project may need extra libraries or dependencies. Utilizing build technologies like Apache Maven or Gradle, one can handle these dependencies. The project's build setup ought to contain the necessarylibraries.

3. Operating system: The project is compatible with Windows, macOS, and Linux for development and execution. It is crucial to make sure the operating system ofchoice is compatible with the Python version being utilized.

4. File System Access: Read and write access to files on the local file system arerequired for the project. Make that the person using the program has the right permissions to access the required directories and files.

5. User interface Components: Additional software components can be needed ifthe project incorporates a graphical user interface (GUI).

6. Reporting and documentation: If the project entails producing reports or documentation, software like Microsoft Office or LaTeX may be needed toefficiently prepare and format the materials.

Sajith Umar

ORIGINALITY REPORT

8%	4%	1%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Walter Sisulu University Student Paper	2%
2	Submitted to BMS College of Engineering Student Paper	1%
3	www.ijsr.net Internet Source	1%
4	www.slideshare.net Internet Source	1%
5	slides.com Internet Source	<1%
6	www.appviewx.com Internet Source	<1%
7	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	<1%
8	www.tsif.com Internet Source	<1%
9	ayoub3bidi.medium.com Internet Source	<1%

10	campus.w3schools.com Internet Source	<1 %
11	www.madonna.org Internet Source	<1 %
12	www.saddleback.edu Internet Source	<1 %
13	Mohammad Akhlaghi, Raul Infante-Sainz, Boudewijn Roukema, Mohammadreza Khellat, David Valls-Gabaud, Roberto Baena Galle. "Towards Long-term and Archivable Reproducibility", Computing in Science & Engineering, 2021 Publication	<1 %
14	sunscrapers.com Internet Source	<1 %
15	www.healthit.gov Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		