1) PROJECT :

Title of the Project: Pre-Care for Diagnosis

1. Abstract :

The "Pre-Care for Diagnosis" application is a Tkinter-based desktop tool designed to streamline the initial patient assessment process by providing an intuitive platform for managing patient data, offering preliminary symptom-based diagnoses, and facilitating data export for record-keeping. Developed with a focus on supporting healthcare providers in triage and outpatient settings, this application enables users to input and store essential patient details such as name, age, and symptoms. Based on the symptoms provided, the application utilizes a predefined set of diagnostic criteria to suggest potential causes, medications, and prescriptions. This allows healthcare providers to make informed decisions more quickly, particularly for cases that require urgent care or further examination.

The application includes features that enhance usability and accessibility, such as a user-friendly interface, clear visual organization, and error handling to ensure accurate data input. A key functionality of the system is its ability to save patient records in an Excel file, facilitating long-term data storage, easy sharing, and integration into broader ealthcare databases. Through structured data export, the application supports healthcare administrators in tracking patient trends, analyzing common symptoms, and identifying prevalent health issues within a population, thereby contributing to healthcare analytics and data-driven decision-making.

Designed to support healthcare providers in resource-constrained environments, the "Pre-Care for Diagnosis" application aims to enhance efficiency in patient data management, promote preventive care through symptom-based recommendations, and reduce administrative workload by automating essential tasks. The flexible diagnostic model also allows for future expansion, enabling customization of symptom-diagnosis mappings as medical knowledge and patient demographics evolve. By meeting these objectives, the "Pre-Care for Diagnosis" application establishes itself as a valuable tool in patient care management, significantly enhancing the quality and speed of initial patient assessments and contributing to a more efficient healthcare delivery system.

2. Objectives of the project:

2.1.Enhance Patient Data Management:

2.1.1:- Provide a centralized system to capture and store essential patient information (e.g., name, age, symptoms) that can be accessed, updated, and stored systematically.

2.1.2:- A structured data repository allows healthcare providers to track patient information efficiently, reducing the risk of data loss and improving administrative workflows.

2. 2. Support Symptom-Based Preliminary Diagnosis:

2.2.1:- Enable the application to offer initial diagnosis suggestions based on symptoms to facilitate quicker, symptom-driven assessments.

2.2.2:- Providing an early hypothesis on possible conditions, medications, and prescriptions enhances the triage process and aids in prioritizing cases that may need immediate attention or further clinical investigation.

2.3. Streamline the Pre-Clinical Process:

2.3.1:- Minimize the time healthcare providers spend on repetitive administrative tasks by automating patient data entry, diagnosis tracking, and symptom-based advice.

2.3.2:-By allowing healthcare providers to focus on critical patient care activities, the application contributes to a more efficient pre-clinical workflow.

2.4. Enable Easy Data Export and Review:

2.4.1:- Facilitate data persistence by exporting patient records to an Excel file, ensuring patient data can be shared, reviewed, and archived for future reference.

2.4.2:-With an exportable format, patient data becomes easier to manage, analyze, and integrate into broader healthcare databases, supporting long-term tracking and audit purposes.

2.5. Provide a User-Friendly and Accessible Interface:

2.5.1:- Develop an intuitive user interface with clear, labeled sections and guidance, making it accessible even to users with minimal technical knowledge.

2.5.2:- A user-friendly design allows healthcare providers to quickly navigate and utilize the application, ensuring that the app can be adopted across different skill levels without extensive training.

2.6. Promote Initial Self-Assessment and Preventive Care Awareness:

2.6.1:-By displaying preliminary diagnoses and relevant medications or preventive measures, raise patient awareness of basic health practices and preventive care.

2.6.2:-Encouraging preventive care advice contributes to patient empowerment and self-care, potentially reducing the frequency of preventable issues and hospital visits.

2.7. Incorporate Expandable and Customizable Diagnostic Criteria:

2.7.1:- Design the diagnosis system to be scalable, allowing for additional symptoms, conditions, or diagnoses to be integrated as medical needs and user requirements evolve.

2.7.2:-A customizable and expandable system provides flexibility, ensuring the app remains relevant as healthcare knowledge advances and new conditions emerge.

2.8.Enhance Record-Keeping for Healthcare Analytics:

2.8.1:- Create a framework for collecting structured patient data that can be used for analytics, trend identification, and quality assessment in healthcare services.

2.8.2:- Structured data collection and export facilitate clinical audits, enabling insights into patient demographics, common symptoms, and prevalent health concerns, which is valuable for healthcare analytics and strategic decision-making.

3. Software and Hardware Requirements:

3.1.Software Requirements:

3.1.1:-Operating System: Windows 7, 8, 10, or later; macOS; or Linux (Ubuntu, Fedora, etc.) Compatible with any OS that supports Python and Tkinter.

3.1.2:-Programming Language: Python 3.7 or above, as the application leverages Python for its GUI (Tkinter), data handling, and logic processing.

3.1.3:-Libraries and Dependencies:

3.1.3.1:-Tkinter: Built-in Python library for creating the graphical user interface.

3.1.3.2:-Pillow (PIL): Required for image processing to display background images in the application.

3.1.3.3:-Openpyxl: Needed for reading from and writing to Excel files for data export and import functionality.

3.1.3.4:-OS Module (Python Standard Library): For file management operations, such as checking if the Excel file exists before writing data.

3.1.3.5:-Optional: IDE (e.g., PyCharm, VS Code, or Jupyter Notebook) for development purposes.

3.1.4:-Excel Software: Microsoft Excel or any other spreadsheet software (like Google Sheets, LibreOffice Calc) for viewing the exported patient data.

3.1.5:- Python Package Manager (pip): Required for installing the necessary libraries, such as `Pillow` and `Openpyxl`, if they are not already installed.

3.1.6:-Database (Optional): For larger-scale or enterprise-level implementation, a relational database like SQLite, MySQL, or PostgreSQL could replace Excel for data storage. However, Excel suffices for smaller-scale applications.

3.2.Hardware Requirements:

3.2.1:-Processor:

3.2.1.1:-**Minimum:** Intel i3 or equivalent for basic functionality.

3.2.1.2:- **Recommended:** Intel i5 or higher (or equivalent) for smoother performance, especially if additional features are integrated.

3.2.2:-Memory (RAM):

3.2.2.1:- **Minimum:** 4 GB to run the application effectively.

3.2.2.2:- **Recommended:** 8 GB or higher, especially if the application is used alongside other software or if future expansions increase data volume.

3.2.3:-Storage:

3.2.3.1:-**Minimum:** 500 MB of free space to accommodate the Python environment, libraries, and Excel files.

3.2.3.2:-**Recommended:** 1 GB or higher, especially if the system is used for prolonged patient record storage and analysis.

3.2.4:-Display:

3.2.4.1:-**Minimum Resolution:** 1024 x 768 for a clear interface display.

3.2.4.2:-**Recommended Resolution:** 1280 x 1024 or higher for better layout presentation, especially for viewing background images and managing multiple windows.

3.2.5:-Other Peripherals:

3.2.5.1:- Mouse and Keyboard: Required for navigating the GUI and entering patient data.

3.2.5.2:-Optional Printer: For printed copies of patient records, if needed.

3.2.5.3:-External Storage (Optional): For data backup and portability, especially if using Excel files that may need external storage.

4. Screenshots of the Project:

4.1:-Welcome Screen:

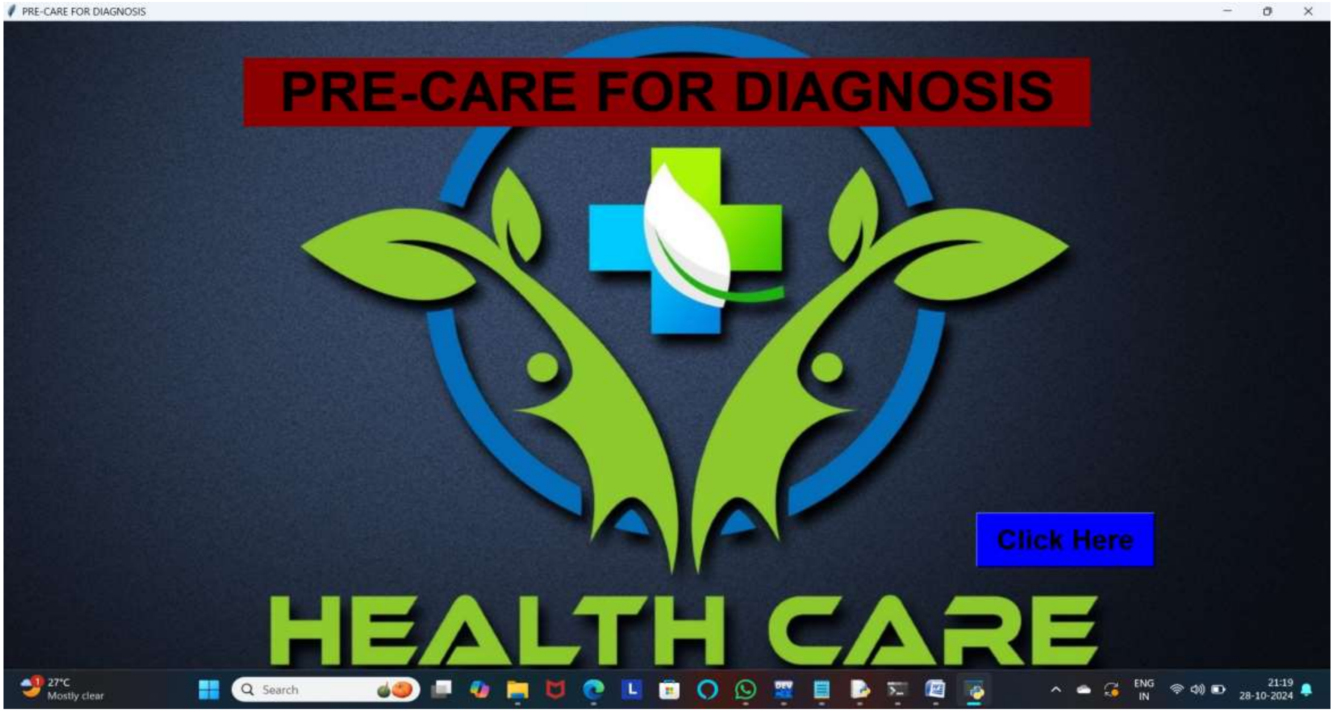


Fig-4.1

4.1.1:- **Description:** Capture the initial welcome screen with the background image, application title ("Pre-Care for Diagnosis"), and the "Click Here" start button. This screen sets the tone for the application's user interface.

4.1.2:-**Details to Highlight:** Background image, title styling, start button placement.

4.2:- Main Screen After Start Button:

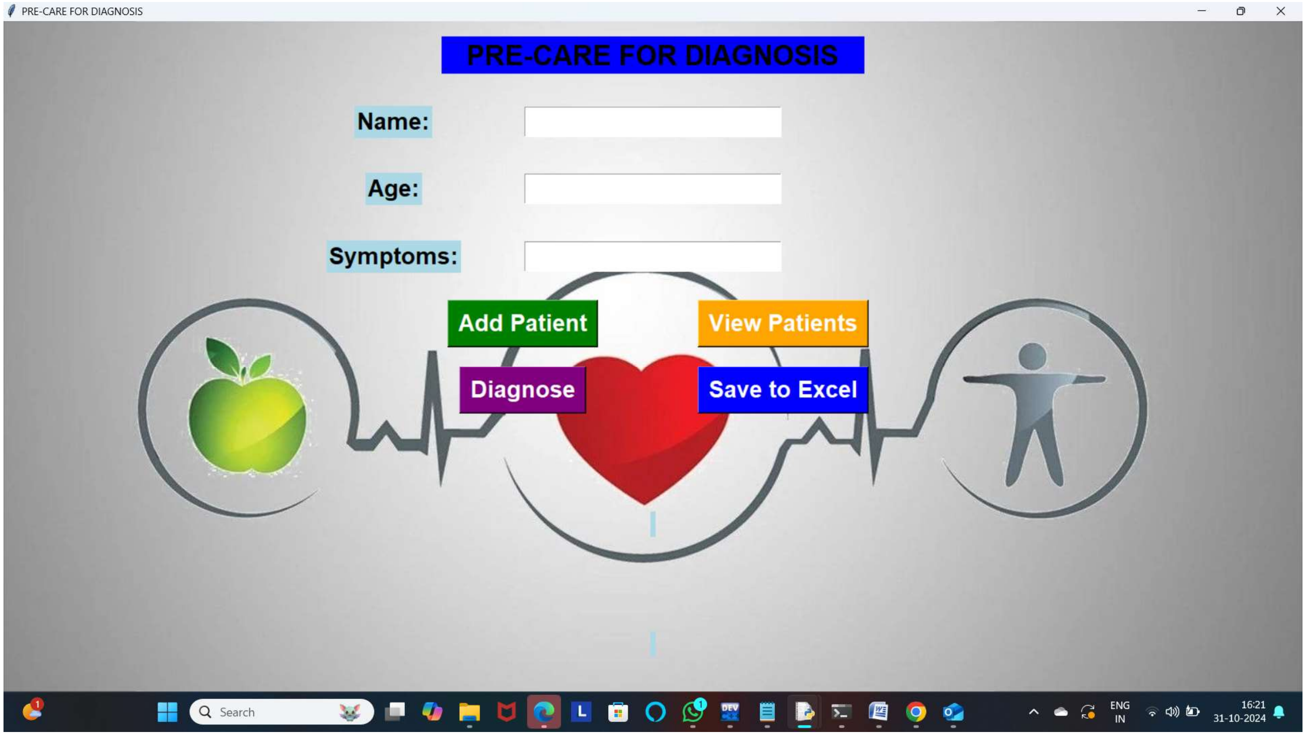
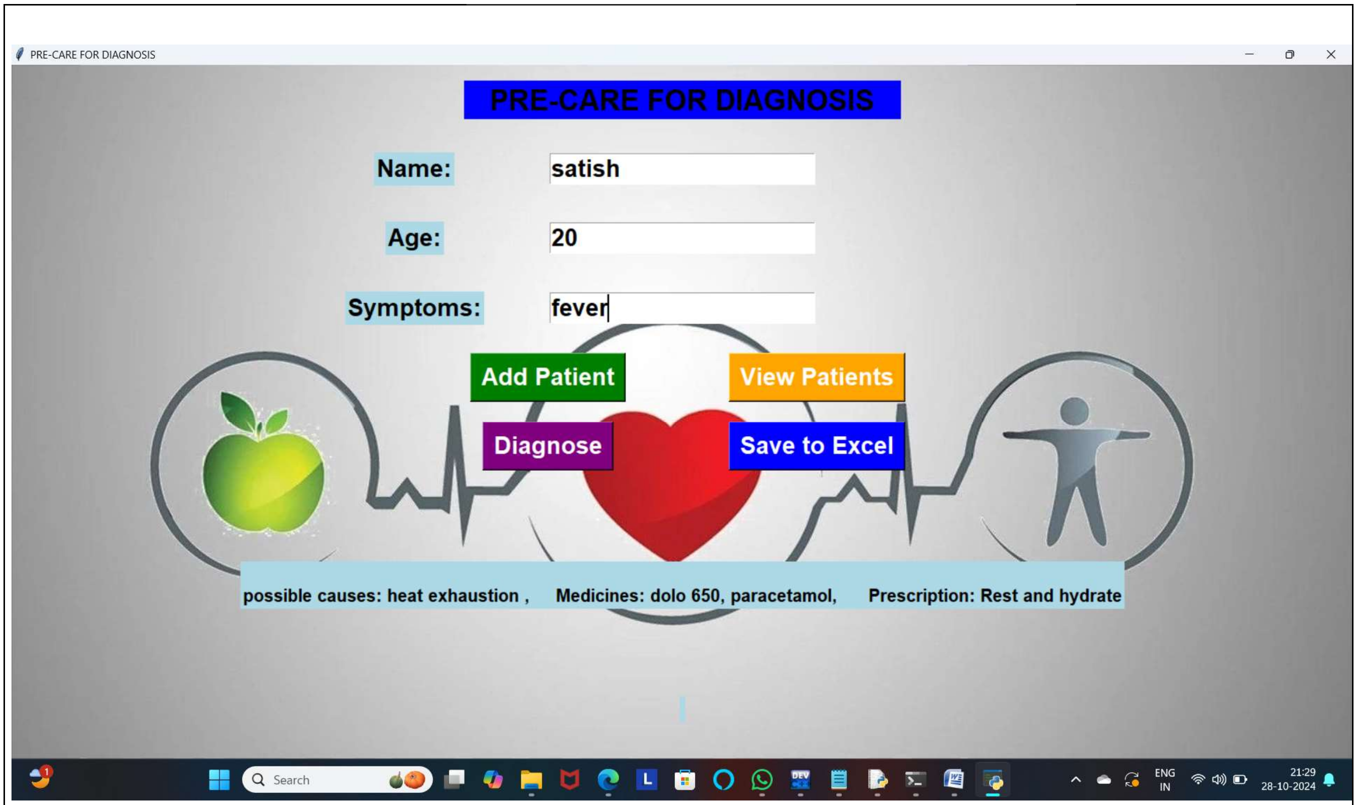


Fig-4.2

4.2.1:- **Description:** Capture the primary user interface that appears after pressing the "Click Here" button. This screen should show input fields for patient name, age, and symptoms, as well as action buttons (Add Patient, View Patients, Diagnose, Save to Excel).

4.2.2:- **Details to Highlight:** Input fields for patient details, action buttons for different functions, background image, title styling, and placement of each element.

4.3:-Diagnose Patient Screen:



PRE-CARE FOR DIAGNOSIS

Name: satish

Age: 20

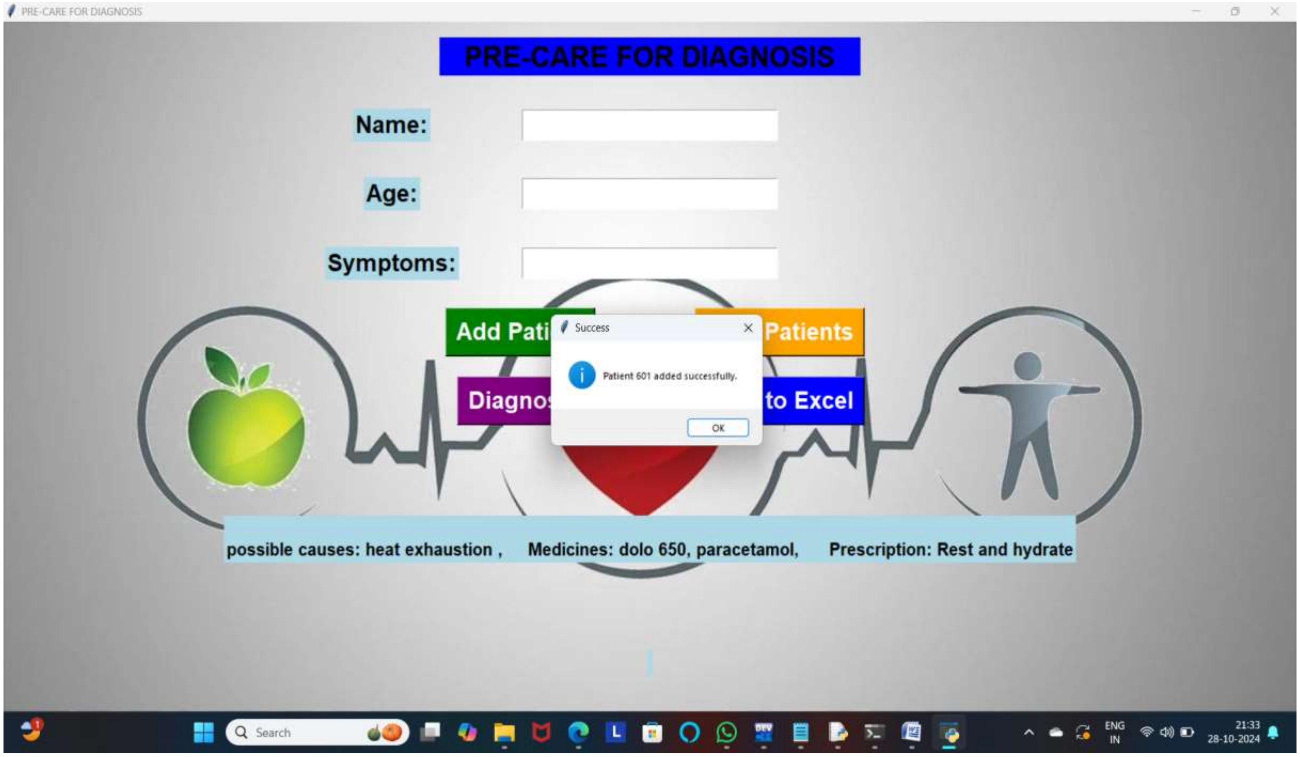
Symptoms: fever

possible causes: heat exhaustion ,   Medicines: dolo 650, paracetamol,   Prescription: Rest and hydrate

4.3.1:- **Description:** Display the output shown after using the "Diagnose" button. It should show the symptom-based preliminary diagnosis, including possible causes, medications, and prescriptions.

4.3.2:-**Details to Highlight**: Diagnosis display area with the full diagnosis, potential causes, suggested medicines, and prescriptions. Ensure readability of text.

4.4:-**Add Patient Confirmation:**



Patient 601 added successfully.

possible causes: heat exhaustion , Medicines: dolo 650, paracetamol, Prescription: Rest and hydrate

Fig-4.4

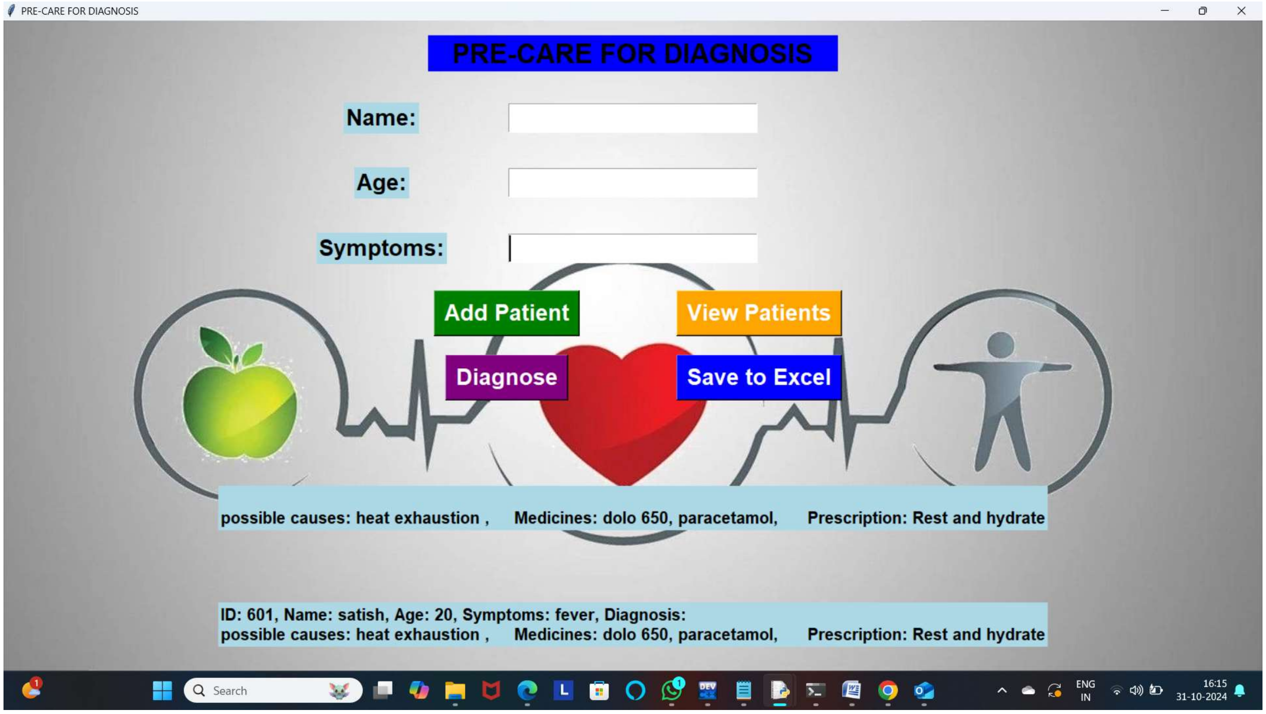
4.4.1:- **Description:** Show the popup confirmation message after adding a patient successfully, indicating that a patient has been added to the system.

4.4.2:- **Details to Highlight:** Success message from messagebox.showinfo, which confirms the patient has been added, and the patient ID displayed.

4.5 :- **View Patients Screen:**

4.5.1:-**Description:** Capture the result displayed after clicking "View Patients." This screen shows a list of all registered patients along with their details (ID, name, age, symptoms, diagnosis).

4.5.2:-**Details to Highlight:** Patient information display area, layout of each patient's data, and styling of the display.

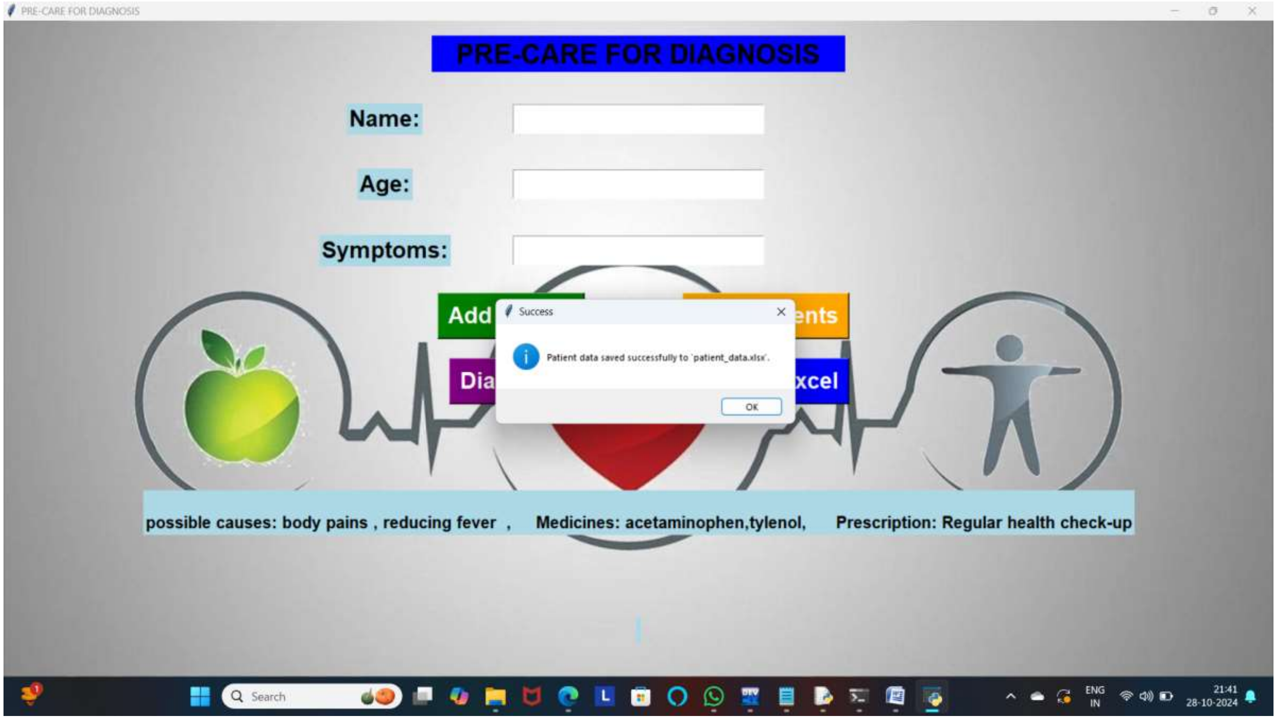


possible causes: heat exhaustion , Medicines: dolo 650, paracetamol, Prescription: Rest and hydrate

ID: 601, Name: satish, Age: 20, Symptoms: fever, Diagnosis:
possible causes: heat exhaustion , Medicines: dolo 650, paracetamol, Prescription: Rest and hydrate

Fig-4.4

4.5 :-Save to Excel Confirmation:



Patient data saved successfully to 'patient_data.xlsx'.

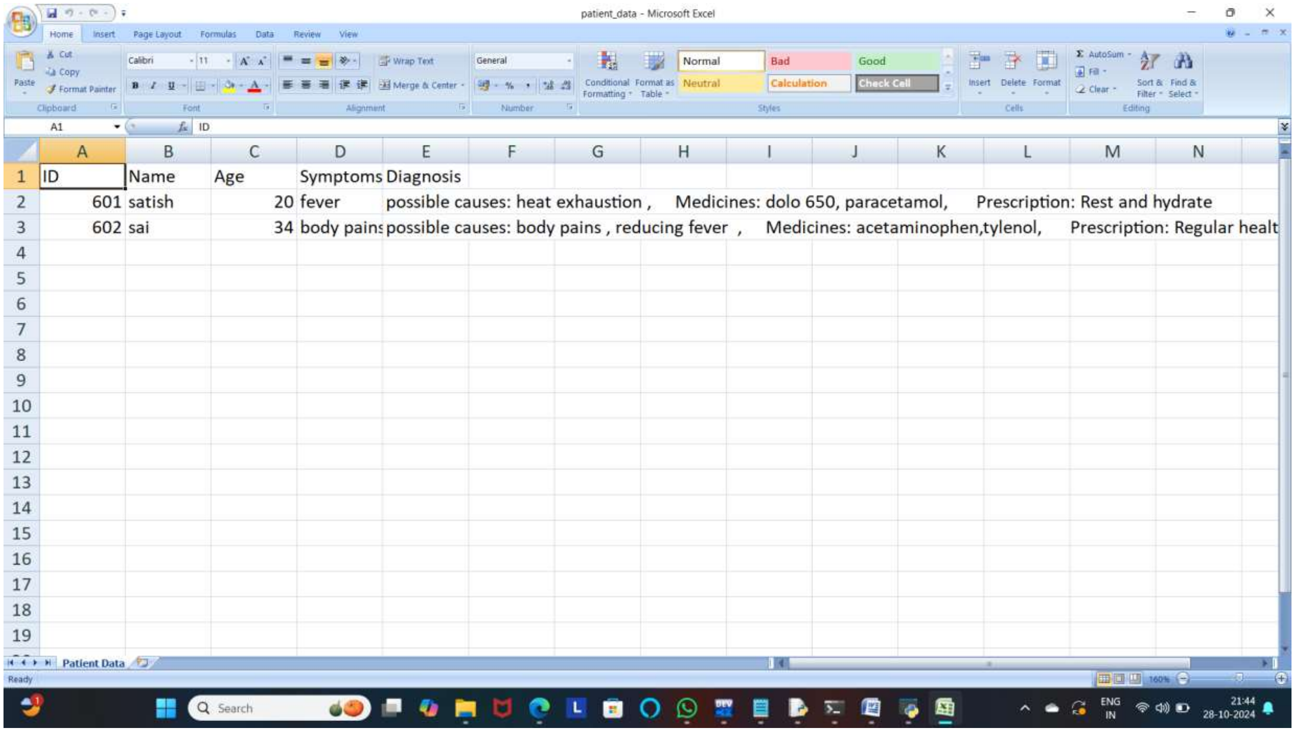
possible causes: body pains , reducing fever , Medicines: acetaminophen,tylenol, Prescription: Regular health check-up

Fig-4.5

4.5.1:-**Description:** Capture the success message shown after saving patient data to the Excel file, verifying the data export functionality.

4.5.2:-**Details to Highlight**: Confirmation popup with the success message indicating the save path of the Excel file.

4.6:-**Excel File with Patient Data:**


patient_data - Microsoft Excel
Home Insert Page Layout Formulas Data Review View
Patient Data
Ready
21:44
28-10-2024

ID	Name	Age	Symptoms	Diagnosis
601	satish	20	fever	possible causes: heat exhaustion , Medicines: dolo 650, paracetamol, Prescription: Rest and hydrate
602	sai	34	body pains	possible causes: body pains , reducing fever , Medicines: acetaminophen,tylenol, Prescription: Regular healt

4.6.1:-**Description:** Open the generated Excel file to show the structured patient data within the spreadsheet, including headers and sample entries.

4.6.2:-**Details to Highlight:** Data organization with headers (“ID,” “Name,” “Age,” “Symptoms,” “Diagnosis”), clarity of information, and layout of the spreadsheet data.

5. Conclusion :

The "Pre-Care for Diagnosis" application is a significant step towards streamlining the preliminary healthcare process by providing a user-friendly, efficient platform for managing patient information and offering symptom-based preliminary assessments. By integrating essential functions such as patient data entry, real-time diagnosis suggestion, and Excel-based data export, the application simplifies administrative and diagnostic workflows, enabling healthcare providers to focus more effectively on patient care.

This project highlights the potential of digital solutions in enhancing healthcare efficiency, particularly in outpatient and triage settings where time and resources are limited. The application's easy-to-use interface and logical workflow make it accessible for healthcare providers with varying levels of technical expertise, while its structured data handling ensures that patient records are organized and accessible. Additionally, the symptom-diagnosis mapping system offers a valuable aid for initial assessments, enabling healthcare professionals to provide timely guidance, recommend preliminary medications, and encourage preventive measures.

The ability to export data to Excel further extends the application's utility by ensuring that patient records can be stored, shared, and reviewed conveniently, supporting broader health data management needs and analytics. This export feature allows the application to serve as a stepping stone toward more comprehensive electronic health record (EHR) systems, particularly in smaller clinics and rural healthcare facilities that may not have access to full-scale EHR solutions.

In conclusion, the "Pre-Care for Diagnosis" application demonstrates how technology can improve healthcare delivery through enhanced data management, preliminary diagnostics, and operational efficiency. Its adaptable structure also allows for future expansion, ensuring it remains relevant as healthcare needs evolve and new diagnostic criteria emerge. With further enhancements, such as the integration of more extensive medical databases or connection to EHRs, the application could expand its role in supporting healthcare providers in both preventive and reactive care, ultimately contributing to improved patient outcomes and a more efficient healthcare ecosystem.