Detection of Endocrine Disorder Based on Machine Learning using Physical and Clinical Database

A project report

submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Technology

in

Computer Science & Engineering

by

Utkarshani Tiwari(2102840100177)

Muskan Singh(2102840100106)

Ashutosh Kumar Singh(2102840100050)

Ashish Singh(2102840100048)

Under the Guidance of

Mr. Prafull Pandey

(Assistant Professor))

Department of Computer Science & Engineering

UNITED INSTITUTE OF TECHNOLOGY PRAYAGRAJ

Uttar Pradesh 211010, INDIA.

(Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow)

2024-2025

Declaration of Academic Ethics

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this project report. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Utkarshani Tiwari)

(Muskan Singh)

(Ashutosh Kumar Singh)

(Ashish Singh)

Certificate from the project guide

This is to certify that the work incorporated in the project report entitled *"Detection of Endocrine Disorder Based on Machine Learning using Physical and Clinical Database"* is a record of work carried out by *Utkarshani Tiwari(2102840100177)*, *Muskan Singh(2102840100106)*, *Ashutosh Kumar Singh(2102840100050)*, *Ashish Singh(2102840100048)*, Under my guidance and supervision for the award of Degree of Bachelor of Technology in Computer Science & Engineering.

To the best of my/our knowledge and belief the project report

1. Embodies the work of the candidates themselves,
2. Has duly been completed,
3. Fulfils the requirement of the Ordinance relating to the Bachelor of Technology degree of the University and
4. Is up to the desired standard both in respect of contents and language for being referred to the examiners.

Date: May 23, 2025 (Prafull Pandey)

The project work as mentioned above is here by being recommended and Forwarded for examination and evaluation.

Date: May 23, 2025 Head of Department

Certificate from External Examiner

This is to certify that the project report entitled *“MY Social Health”*, submitted by:

- *Utkarshani Tiwari (2102840100177)*
- *Muskan Singh (2102840100106)*
- *Ashutosh Kumar Singh (2102840100050)*
- *Ashish Singh (2102840100048)*

has been examined by the undersigned as a part of the examination for the award of the degree of *Bachelor of Technology in Computer Science & Engineering*.

Internal Examiner

Date: ____________

External Examiner

Date: ____________

Acknowledgments

I, along with my team, would like to convey our heartfelt gratitude to **Dr. Amit Kumar Tiwari** and **Mr. Prafull Pandey** for their valuable support and technical suggestions throughout the project.

I am deeply grateful to Anshu, Hardik, and Utkarsh for their unconditional support during the course of this work.

I would also like to express my sincere thanks to all the members of the Computer Science and Engineering department for their direct or indirect contributions to this project, and for providing us with a friendly and supportive environment in the lab.

Finally, I extend my deepest gratitude to my family members and all my friends for their constant moral support.

(Utkarshani Tiwari)

(Muskan Singh)

(Ashutosh Kumar Singh)

(Ashish Singh)

Abstract

Polycystic Ovary Syndrome (PCOS) is a prevalent and complex endocrine disorder that affects approximately 5–10 of women of reproductive age. It is primarily characterized by hormonal imbalances, irregular or absent ovulation, menstrual irregularities, and the presence of multiple cysts in the ovaries. Despite its widespread occurrence, PCOS often remains underdiagnosed due to the variability in symptoms, which can overlap with other conditions such as thyroid disorders or insulin resistance. As a result, many women experience delayed diagnosis and treatment, increasing their risk of complications like infertility, obesity, type 2 diabetes, cardiovascular disease, and mental health concerns. Early and accurate detection of PCOS is critical to managing the condition and reducing associated health risks. Traditional diagnostic methods include clinical evaluation, ultrasonography, and hormonal assays. However, these approaches have limitations, particularly in cases where symptoms are subtle or atypical. Additionally, the lack of standardized diagnostic criteria contributes to inconsistent identification and treatment plans. With advancements in healthcare technology, new methods are emerging to improve diagnostic accuracy. Machine learning (ML) and artificial intelligence (AI) have shown significant promise in enhancing PCOS detection by analyzing complex medical data. These technologies can identify patterns and correlations that are not easily visible through traditional techniques. AI-driven models, trained on clinical and biochemical datasets, have the potential to support healthcare providers in making faster and more reliable diagnoses.

This report explores both conventional and AI-assisted diagnostic strategies, evaluating their strengths and limitations. While challenges such as data quality, privacy concerns, and the need for clinical validation remain, the application of AI in PCOS diagnosis holds great promise.

In conclusion, blending traditional diagnostic expertise with advanced computational tools can

revolutionize the way PCOS is detected and managed. This approach not only enhances diagnostic precision but also paves the way for more efficient, patient-centered care for women affected by this condition.

Contents

Abstract **5**

List of Figures **9**

1 Introduction **10**

2 SDLC Model Used **12**

2.1 Requirements Gathering Phase 15

2.2 Design Phase 15

2.2.1 Use Case Diagram 15

2.2.2 Entity Relationship Diagram 15

2.2.3 Data Flow Diagram 15

2.3 Development Phase 15

2.4 Testing Phase 15

2.4.1 Unit Test 15

2.4.2 Medium Test 16

2.4.3 End to End Test 17

3 Requirement Gathering **19**

3.0.1 User System Requirements 20

3.0.2 Devloper's System Requirements 20

4 Design **21**

4.0.1 Use Case Diagram 21

4.0.2 Entity Relationship Diagram 23
4.0.3 Data Flow Diagram 24
4.0.4 Gantt chart 27

5 Implementation **29**

6 Testing **38**

7 Conclusion and Future Work **42**
7.1 Conclusion 42
7.2 Future Work 43

List of Figures

2.1 Phases of Iterative (The Software Development Life Cycle: A Complete Guide) 14

4.1 Use Case Diagram 22
4.2 ER Diagram 24
4.3 Zero level DFD 25
4.4 One level DFD 26
4.5 Gantt Chart 28

5.1 Login Page 30
5.2 Registration Page 31
5.3 Home Page 32
5.4 User Input Page 33
5.5 User Input Page 34
5.6 User Input Page 35
5.7 Diet Chart 36
5.8 Exercise Chart 37

6.1 Unit Testing 39
6.2 Integration Testing 40
6.3 System Testing 41

Chapter 1

Introduction

Polycystic Ovary Syndrome (PCOS) is among the most widespread endocrine disorders affecting women of reproductive age, impacting an estimated 5–10 One of the key issues with PCOS is that its symptoms can be broad and may mimic those of other health conditions, such as thyroid disorders, insulin resistance, or adrenal hyperplasia. Symptoms commonly include acne, weight gain, excessive hair growth, and infertility. Because of this overlap, many cases remain undiagnosed or are only identified after complications arise. Without proper management, PCOS can lead to long-term health issues, including type 2 diabetes, metabolic syndrome, infertility, cardiovascular problems, and mental health concerns such as depression and anxiety. [**?**]
Early and accurate diagnosis of PCOS is crucial in preventing these complications and enabling effective treatment plans. Currently, diagnosis typically involves a combination of clinical evaluations, pelvic ultrasonography to detect ovarian cysts, and hormonal blood tests to assess androgen and insulin levels. However, these conventional methods have their limitations, particularly due to the heterogeneous nature of the syndrome.
To improve diagnostic accuracy and efficiency, this report explores the growing role of advanced technologies in PCOS detection. Specifically, it investigates the use of machine learning algorithms and artificial intelligence (AI)-driven models, which have shown great promise in analyzing complex datasets. These technologies can detect patterns and correlations in medical data that might be missed through traditional diagnostic approaches. [**?**]
The study also highlights the challenges of integrating AI into clinical practice, such as ensuring

data privacy, managing variations in patient data, and the need for physician training. Despite these obstacles, the potential benefits are significant. AI-enhanced diagnostic tools can not only improve detection rates but also help create personalized treatment plans, reduce diagnostic delays, and lower overall healthcare costs.

In conclusion, while traditional diagnostic methods remain essential, the incorporation of AI and data-driven models could greatly enhance the detection and management of PCOS, leading to better patient outcomes and a more proactive approach to women's health. [**?**]

Chapter 2

SDLC Model Used

For the development of our project, **"PCOS Detection Using Machine Learning"**, we adopted the **Iterative Software Development Life Cycle (SDLC)** model. The iterative model is highly suitable for projects where requirements are expected to evolve and where the solution benefits from progressive refinements—as is typical in data-driven and machine learning applications. Unlike the linear Waterfall model, which proceeds in a strict sequence, the Iterative model allows repeated cycles of development. Each iteration involves planning, design, implementation, and evaluation, helping teams adapt quickly to new insights and feedback. This aligns well with our project, which required experimenting with different machine learning algorithms, tuning hyperparameters, and refining results based on model performance.

Why Iterative Model?

The Iterative model was chosen for the following reasons:

- **Evolving Requirements:** Our understanding of the data and the problem space deepened as the project progressed.
- **Experimentation with Algorithms:** The need to try multiple ML models (e.g., Logistic Regression, SVM) and performance metrics encouraged an iterative workflow.
- **Early Testing and Feedback:** It allowed us to test early versions of the model and incorporate suggestions from academic mentors and healthcare domain experts.

Phases of Our Iterative Development

The development of the project progressed through the following iterations:

- **Iteration 1: Requirement Gathering and Initial Design**
  Identified the core objective—early detection of PCOS using patient data. We collected available datasets, understood features, and outlined the machine learning pipeline.

- **Iteration 2: Data Preprocessing and Feature Engineering**
  Handled missing values, encoded categorical features, and normalized numerical attributes. Initial statistical analysis was performed to understand feature distributions.

- **Iteration 3: Model Selection and Initial Training**
  Implemented multiple models (Logistic Regression, SVM) and trained them on a split of the data. Basic performance metrics were generated to compare early results.

- **Iteration 4: Model Tuning and Cross-Validation**
  Performed hyperparameter tuning and applied k-fold cross-validation. Feature importance was analyzed and adjustments were made to improve precision and recall.

- **Iteration 5: Result Visualization and Documentation**
  Created visualizations like confusion matrices, ROC curves, and bar charts of feature importance. This iteration also included writing detailed documentation and preparing the model for possible deployment.

Each iteration provided valuable feedback, allowed us to identify shortcomings early, and reduced overall development risk.

Benefits Observed

- Enabled flexible development through incremental improvements.
- Allowed continuous testing and evaluation, increasing reliability.
- Facilitated better collaboration and faster identification of issues.

- Enhanced the model's overall performance through repeated refinements.

Conclusion

In summary, the Iterative SDLC model provided a structured yet adaptable framework that supported experimental and data-driven development. By allowing multiple cycles of improvement, it contributed significantly to building a more accurate, reliable, and user-oriented machine learning solution for early PCOS detection.

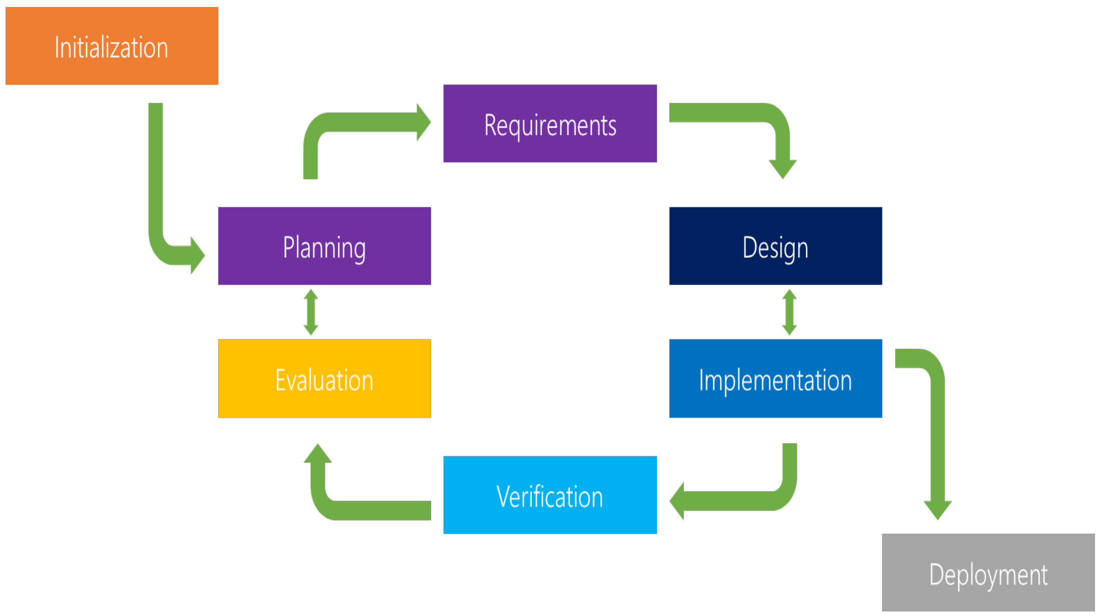


Figure 2.1: Phases of Iterative (The Software Development Life Cycle: A Complete Guide)

2.1 Requirements Gathering Phase

During this phase all the analysis of market and user requirement related to our project is carried out. factors such as cost management , users availability , location check , public needs etc all these points are kept in mind while gathering of requirements.

2.2 Design Phase

During this phase the external view of the system i.e the documented view of our project is created , which include various diagrams representing our project and its features. some of them are:

2.2.1 Use Case Diagram

2.2.2 Entity Relationship Diagram

2.2.3 Data Flow Diagram

2.3 Development Phase

During this phase the step by step development of our project is shown. development in our term means , the change in structure and quality of project with respect to time.

2.4 Testing Phase

Testing phase states that the project which is being deployed has to be tested on certain defined standards to withstand high quality result.some of thise testings are:

2.4.1 Unit Test

Unit testing involves verifying individual components or functions of the software in isolation to ensure that each part behaves as expected. In the context of our PCOS Detection Using

Machine Learning project, unit tests were crucial to validate the integrity of specific modules in our codebase before they were integrated into the complete system.

Key areas covered during unit testing:

- **Data Preprocessing Functions:** We tested functions responsible for handling missing values, normalization, and encoding to ensure they operated correctly and consistently without introducing data errors.
- **Feature Engineering Methods:** Unit tests ensured the logic used to create or transform features (e.g., BMI calculation, categorical encoding) was implemented correctly and returned expected outputs.
- **Machine Learning Model Initialization:** Each machine learning algorithm (e.g., Logistic Regression, SVM) was tested for proper initialization and compatibility with the dataset shape and structure.
- **Evaluation Metric Calculations:** Functions that calculate accuracy, precision, recall, and F1-score were tested with known inputs to ensure the computed results were accurate.
- **Data Loading and Splitting:** Unit tests confirmed that the dataset was loaded correctly and split properly into training and testing subsets without data leakage.

Python's `unittest` and `pytest` frameworks were used to automate and execute these tests. By running unit tests at regular intervals, we were able to detect bugs early in the development process and ensure the reliability of critical components before moving to the integration or system testing phases.

2.4.2 Medium Test

Medium testing, also known as integration testing, focuses on verifying the interactions between multiple modules or components that have already passed unit testing. The goal is to ensure that these components work together as expected when integrated.

In our PCOS Detection Using Machine Learning project, medium testing was essential to validate the combined behavior of connected modules such as data preprocessing, model training, and result evaluation.

Key areas tested during medium testing:

- **Data Pipeline Integration:** Verified the seamless flow of data from the raw dataset through preprocessing, feature selection, and into the machine learning models.
- **Model and Evaluation Compatibility:** Ensured that the trained models generated predictions in the correct format and that these predictions could be evaluated using standard performance metrics (accuracy, precision, recall, F1-score).
- **Error Handling Between Modules:** Tested the system's ability to handle unexpected values or data inconsistencies between modules without crashing, such as missing features or type mismatches.
- **Cross-Validation Logic:** Checked the integration of cross-validation methods to ensure model robustness and generalization when applied to different data splits.

Medium tests provided confidence that independently working components could collaborate effectively within the system. These tests were executed manually and semi-automatically using custom test scripts, ensuring functional correctness and stability across connected parts of the machine learning pipeline.

2.4.3 End to End Test

End-to-End (E2E) testing is the final phase of the testing process where the complete flow of the application is tested from start to finish. The goal is to simulate real-world scenarios and ensure that the entire system works as intended when all modules and components are integrated.

In our project, PCOS Detection Using Machine Learning, end-to-end testing validated the full functionality of the system, starting from raw data input to the final model prediction and result interpretation.

Key areas covered during end-to-end testing:

- **User Data Input Simulation:** Tested the system using sample patient data to ensure proper input handling and compatibility with preprocessing functions.
- **Complete Workflow Execution:** Validated the full pipeline, including data cleaning, feature selection, model training, prediction, and evaluation—executed in the correct sequence without errors.
- **Performance Evaluation Output:** Verified that the performance metrics (accuracy, precision, recall, F1-score) generated at the end matched expectations based on the model and test dataset.
- **Result Visualization:** Ensured that the graphical outputs, such as confusion matrices and performance plots, were generated correctly and accurately represented model outcomes.
- **Robustness and Edge Cases:** Simulated edge cases like incomplete records or out-of-range values to confirm that the system gracefully handled them without crashing.

End-to-end testing confirmed that the entire system was production-ready and reliable for real-world use. It also ensured that any interdependencies between modules were correctly managed, and the project met its original objectives in terms of functionality, accuracy, and usability.

Chapter 3

Requirement Gathering

The requirements for our PCOS Detection System were collected through extensive research, surveys, and user feedback. These requirements were documented in simple, user-friendly language to ensure clarity for both developers and potential users. During this phase, we identified several potential challenges, advantages, and limitations of the project, allowing us to refine our goals and explore alternate solutions effectively.

To assess the success potential of our system, we conducted thorough market research using online sources. Given the competitive nature of the health-tech domain, we evaluated existing applications and identified gaps in their offerings. This helped us understand user expectations, recurring issues, and areas where competitors have struggled—especially those that led to user dissatisfaction.

Surveys focused on what features users desire in upcoming health applications, such as accuracy, user privacy, ease of use, and personalized recommendations. These insights directly influenced the development of our features and planning of system models.

We also studied leading health platforms like Fitbit, Strava, and Headspace, analyzing their business models, user engagement strategies, and technical strengths and weaknesses. These case studies helped us shape a system that is both functional and user-centric.

Our requirement analysis is dynamic and will continue to evolve as new modules are added or user needs change, ensuring our system stays relevant and impactful in the healthcare domain.

3.0.1 User System Requirements

.

- $Software\ Requirements$:
- Android Version should be Kitkat 4.4 or above
- $Hardware\ Requirements$:
- 2 GB RAM
- 1.3 GHz Processor
- 16 GB ROM

3.0.2 Devloper's System Requirements

.

- $Software\ Requirements$:
- Android Studio 6.0
- windows 11
- $Hardware\ Requirements$:
- 8 GB RAM
- Ryzen 5 octa core Processor
- 512 GB ROM

Chapter 4

Design

In the design phase of our PCOS detection project, we focused on clearly defining the system's structure and functionality. This phase is essential for transforming the requirements into a working model. To help with the design, we conducted surveys and analysis to ensure the system meets user expectations. We also used various notational diagrams to map out the system's architecture, data flow, and interactions, ensuring clarity and precision in the development process

4.0.1 Use Case Diagram

A Use Case Diagram is a crucial component of system modeling that helps in externally visualizing the functional behavior of the system. It depicts how different types of users (called actors) interact with the system to perform specific tasks, called use cases. The diagram serves as a blueprint to represent the dynamic nature of the system, particularly how it responds to changing requirements and user inputs. It binds together the functionality and services of the system, showcasing the essential operations required for the successful execution of a platform or application.

In our project, PCOS Detection based on Machine Learning, the Use Case Diagram helps in illustrating the interaction between the system and its two main actors: Patient and Healthcare Provider. These actors interact with the system differently, depending on their roles and requirements.

The Patient uses the system to input personal and medical details such as age, weight, menstrual cycle history, hormone levels, and other relevant health data. Patients can also upload medical test results, like ultrasound or blood tests. Once the data is submitted, the system processes it using machine learning algorithms to predict the likelihood of PCOS. Patients can then view their diagnosis and download reports for further consultation. The Healthcare Provider accesses patient diagnosis reports, reviews historical health data, and may provide treatment plans or recommend additional tests. This role focuses on analysis and decision-making based on the system's predictions.

The primary purpose of the Use Case Diagram in this project is to portray the functional scope of the system and demonstrate how users interact with it in a structured and meaningful way.

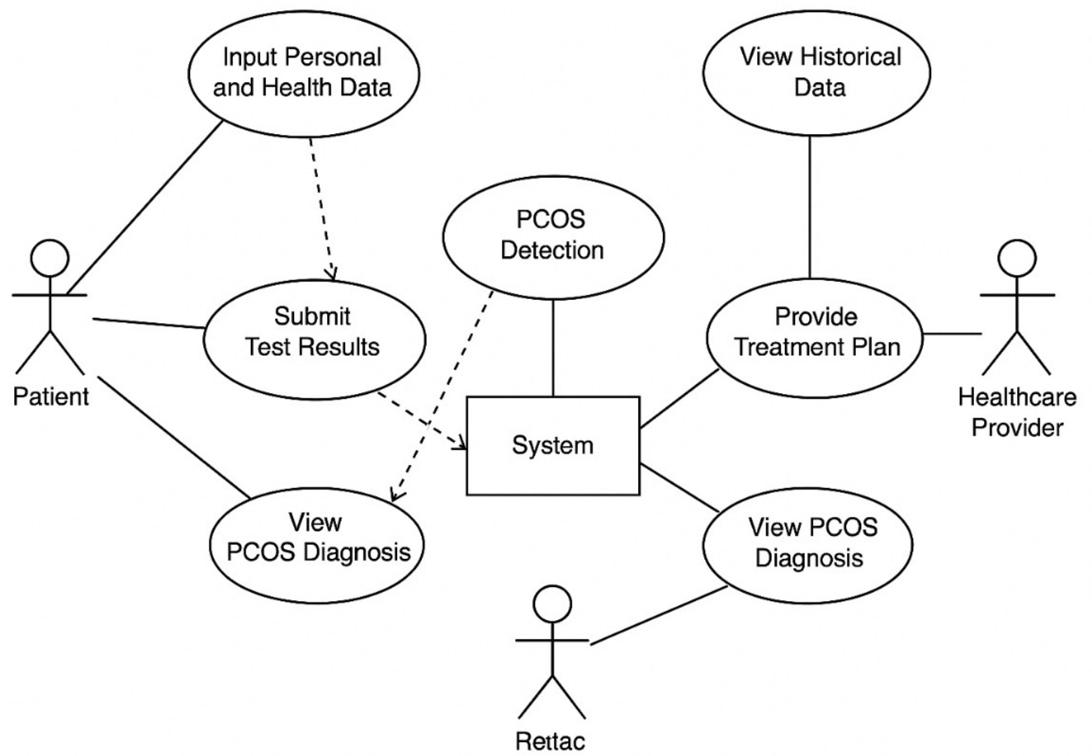


Figure 4.1: Use Case Diagram

4.0.2 Entity Relationship Diagram

An Entity Relationship Diagram (ERD) is a visual tool used in database design to represent the key data entities and the relationships between them. It helps structure how information is stored, accessed, and linked, ensuring efficient and organized data management in any system. In ERDs, rectangles represent entities, ovals represent attributes, and diamonds show relationships among entities.

In our PCOS Detection using Machine Learning project, the ERD defines the interaction between essential components like Patient, Test Result, Diagnosis, and Authentication.

The Patient entity stores user details such as name, age, and health history. The Test Result entity holds medical reports like hormone levels or ultrasound findings submitted by the patient. The Diagnosis entity contains results generated by the machine learning model indicating the likelihood of PCOS. The Authentication entity handles user login credentials for system access. These entities are interrelated—for example, one patient may have multiple test results and diagnoses. The ERD supports the system's backend by clearly modeling these relationships and ensuring data integrity. Overall, the ERD helps visualize how data flows and supports the accurate and secure functioning of the PCOS detection system.

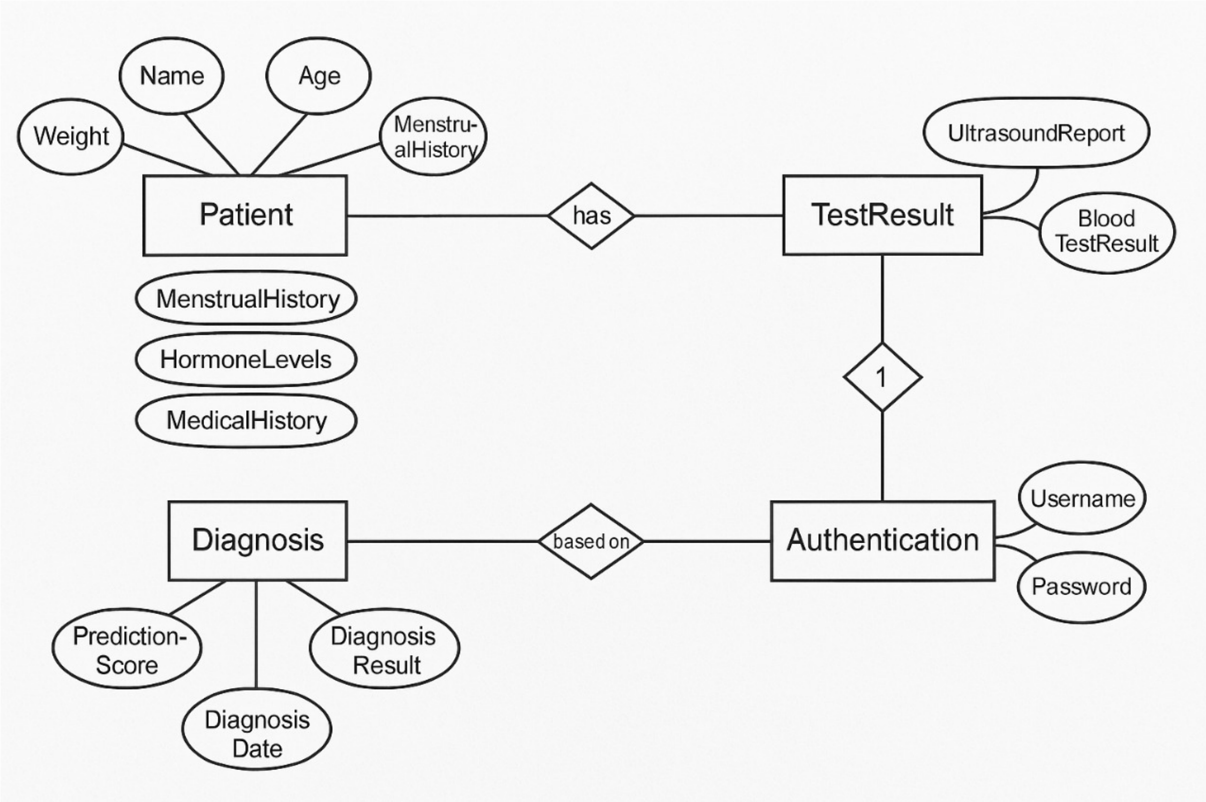


Figure 4.2: ER Diagram

4.0.3 Data Flow Diagram

A Data Flow Diagram (DFD) represents how data moves through the PCOS Detection System. It shows the flow of patient information from input to diagnosis using machine learning. The DFD helps visualize how data is processed, stored, and transformed at each stage. It identifies the interaction between users, system components, and data storage. This helps developers understand how the system operates internally for accurate PCOS prediction.

Data Flow Diagram Layers

- **0-Level DFD**

  A Data Flow Diagram (DFD) shows how data moves through the PCOS Detection System, from user input to final diagnosis. It highlights the processes, data storage, and interactions between patients, healthcare providers, and the system. The DFD helps visualize internal data handling for accurate PCOS prediction.

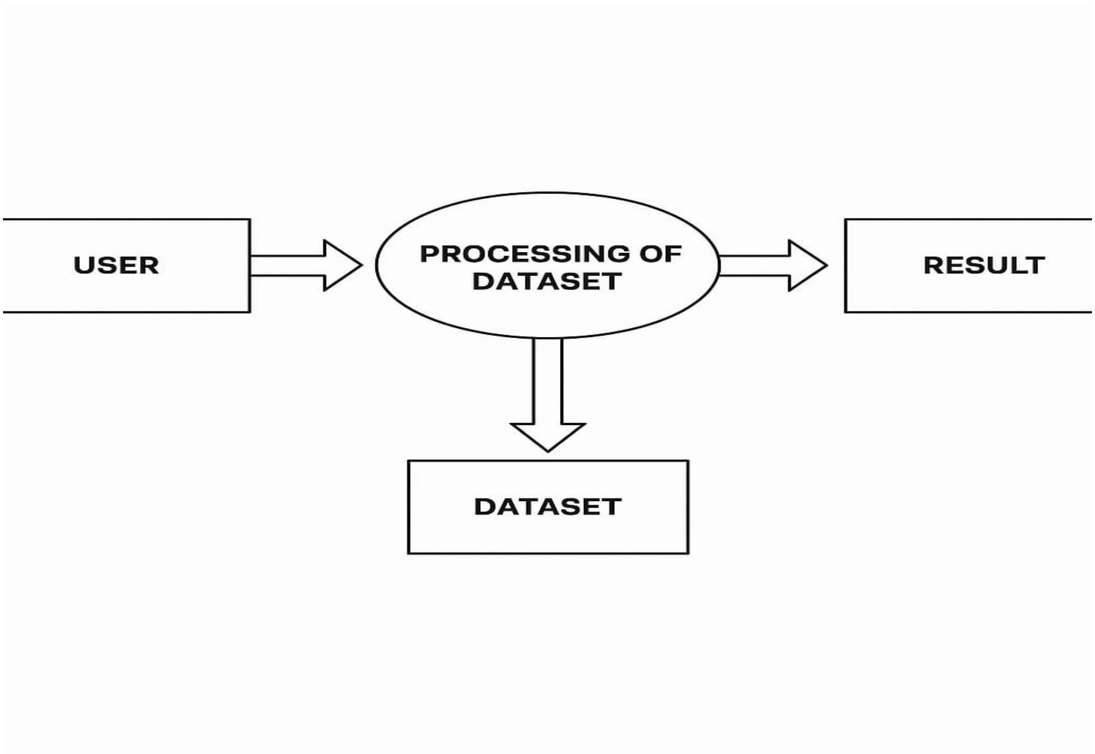


Figure 4.3: Zero level DFD

- **1-Level DFD**

The Level 1 Data Flow Diagram (DFD) expands upon the high-level processes shown in Level 0 by breaking them down into smaller, more detailed sub-processes. At this stage, we highlight the core functionalities of the PCOS Detection System and decompose the main process into meaningful tasks. In our project, the Level 1 DFD illustrates essential features such as Input Health Data, Upload Test Results, Run PCOS Prediction, View Diagnosis, and Generate Reports. It also shows how these processes interact with internal databases like Patient Data, Test Results, and Diagnosis Reports. This diagram helps visualize how user inputs are processed and what outputs are produced, making it easier to understand the flow and structure of the system.

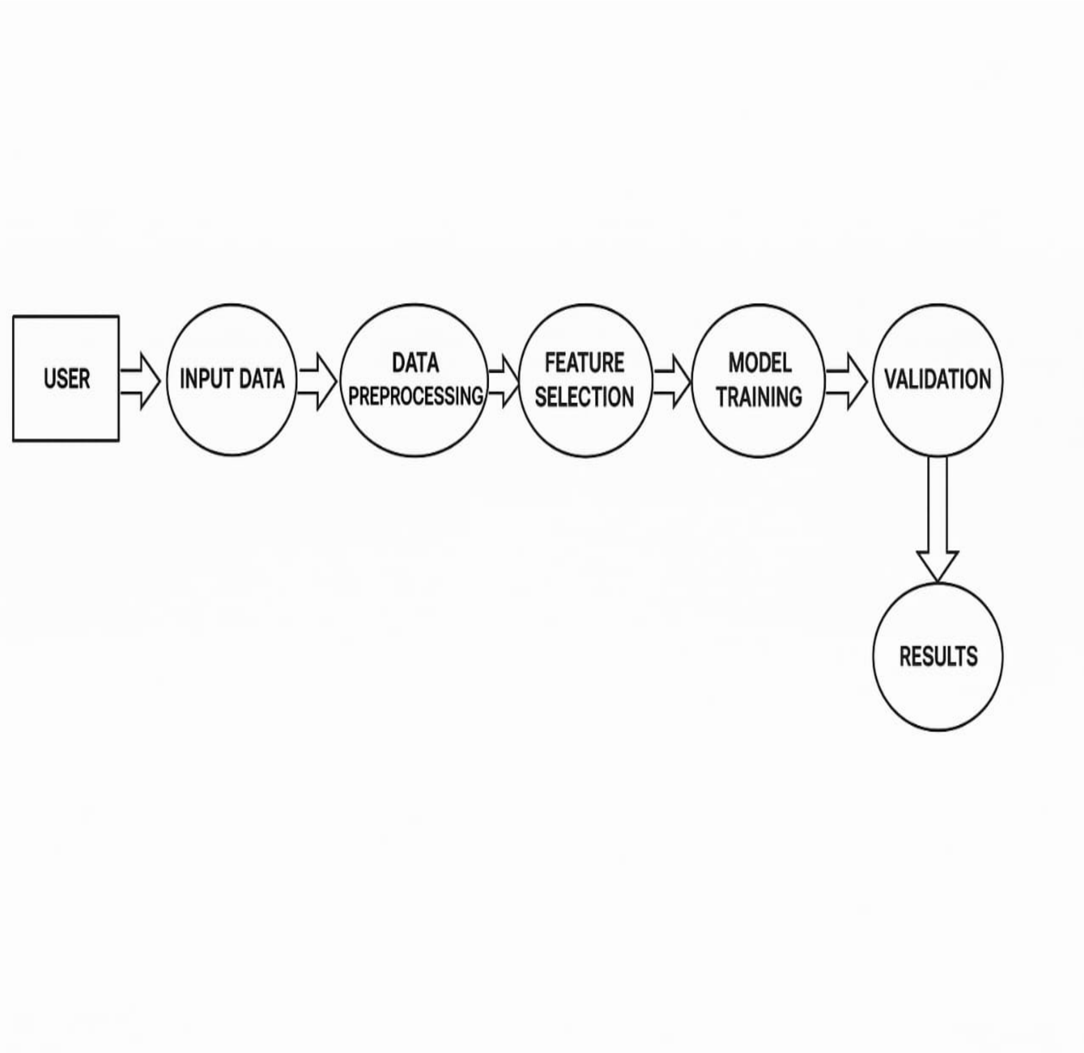


Figure 4.4: One level DFD

4.0.4 Gantt chart

The Gantt chart provides a visual timeline of how we approached the development of our PCOS Detection System. It highlights each phase of the project lifecycle, including planning, designing, implementation, and follow-up, and shows how long each stage took. This chart serves as a roadmap, helping us track progress and manage time effectively.

For our project, we adopted the Agile development model, which supports parallel execution of tasks. Unlike traditional models where one phase must end before the next begins, Agile allows for overlapping activities—enabling continuous planning, design, and development.

In the first iteration, planning began in mid-September and continued into mid-October. Alongside, we carried out in-depth research on existing health applications, analyzing their drawbacks to better define our features. During the research phase, we also initiated the design of our system.

The second iteration involved planning and designing new modules to enhance system capabilities beyond existing solutions. Implementation of these modules began alongside their design, maintaining Agile's flexibility. During development, we identified areas for improvement, returned to the planning phase, and revised the designs before resuming implementation.

After successful implementation, we entered the follow-up phase, collecting user feedback. Based on this input, we continue refining the system to ensure user satisfaction and improve future outcomes. The Gantt chart effectively illustrates this iterative and dynamic development process.

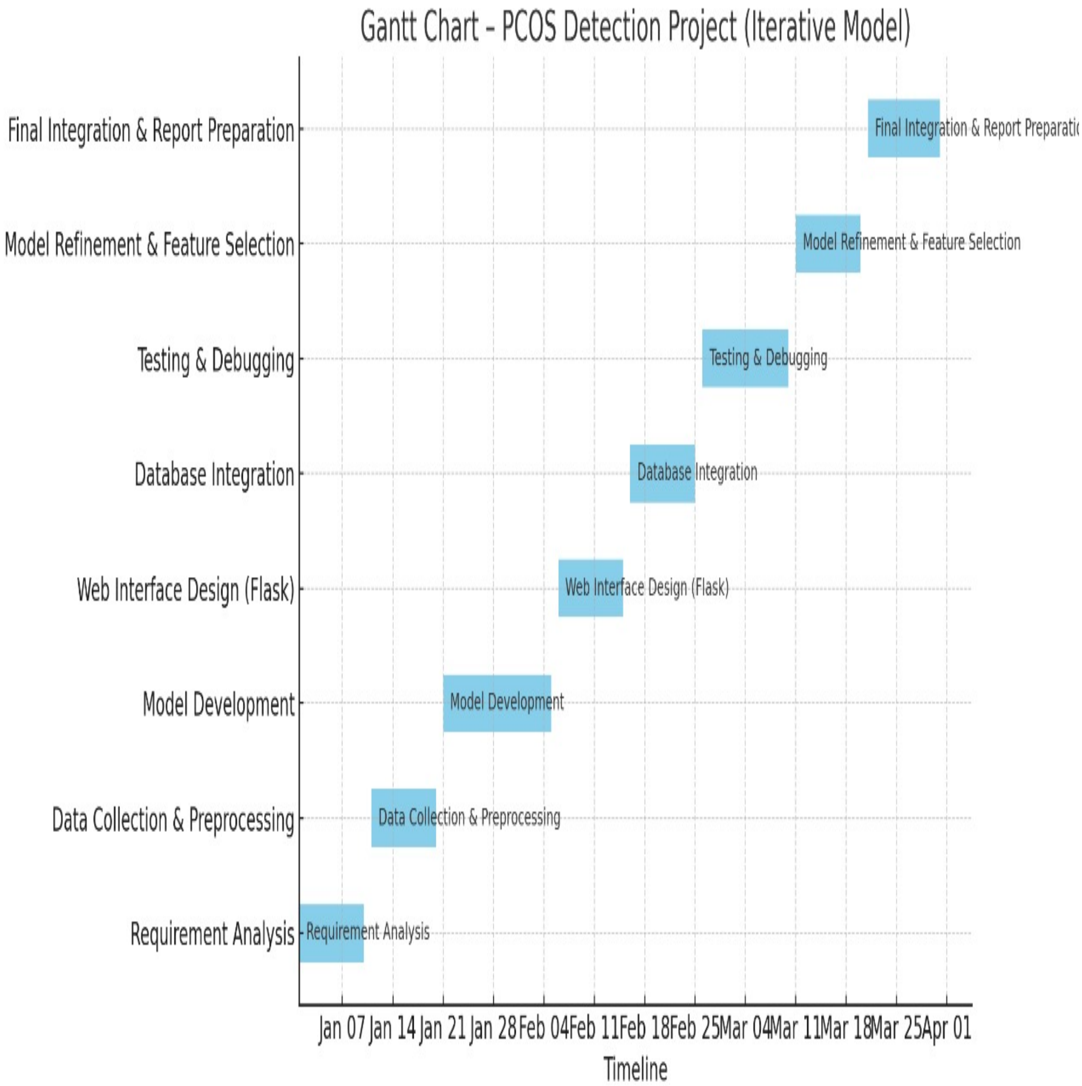


Figure 4.5: Gantt Chart

Chapter 5

Implementation

The development of our PCOS Detection Application commenced after aligning the goals between developers and users. The application was built modularly, with each phase focused on refining functionality and integrating advanced features for better prediction accuracy.

Every module—ranging from data input to diagnosis using machine learning—was thoroughly tested before deployment. The final version underwent complete system testing to ensure real-world reliability. Each module performs distinct tasks independently, contributing to accurate PCOS analysis.

Implementation began with team collaboration, task delegation, and building the base model. This included UI/UX design, ML model integration, and smooth navigation for users. Due to its complexity, the project required strong coordination and time investment.

As user needs evolved, the prototype was continuously improved. Developers adhered to coding standards and employed efficient tools like compilers and debuggers. All technologies and languages were selected to match project requirements, ensuring a scalable, reliable, and user-friendly solution.

The implementation also focused on maintaining data privacy and security for health-related information. Feedback from initial users helped shape refinements during development. This iterative approach ensured the final product not only met functional goals but also delivered a smooth and effective user experience.

- This is the Login Page of our PCOS Detection application, which appears first after launching the app. Users can log in using their mobile number to access personalized health prediction and analysis features.

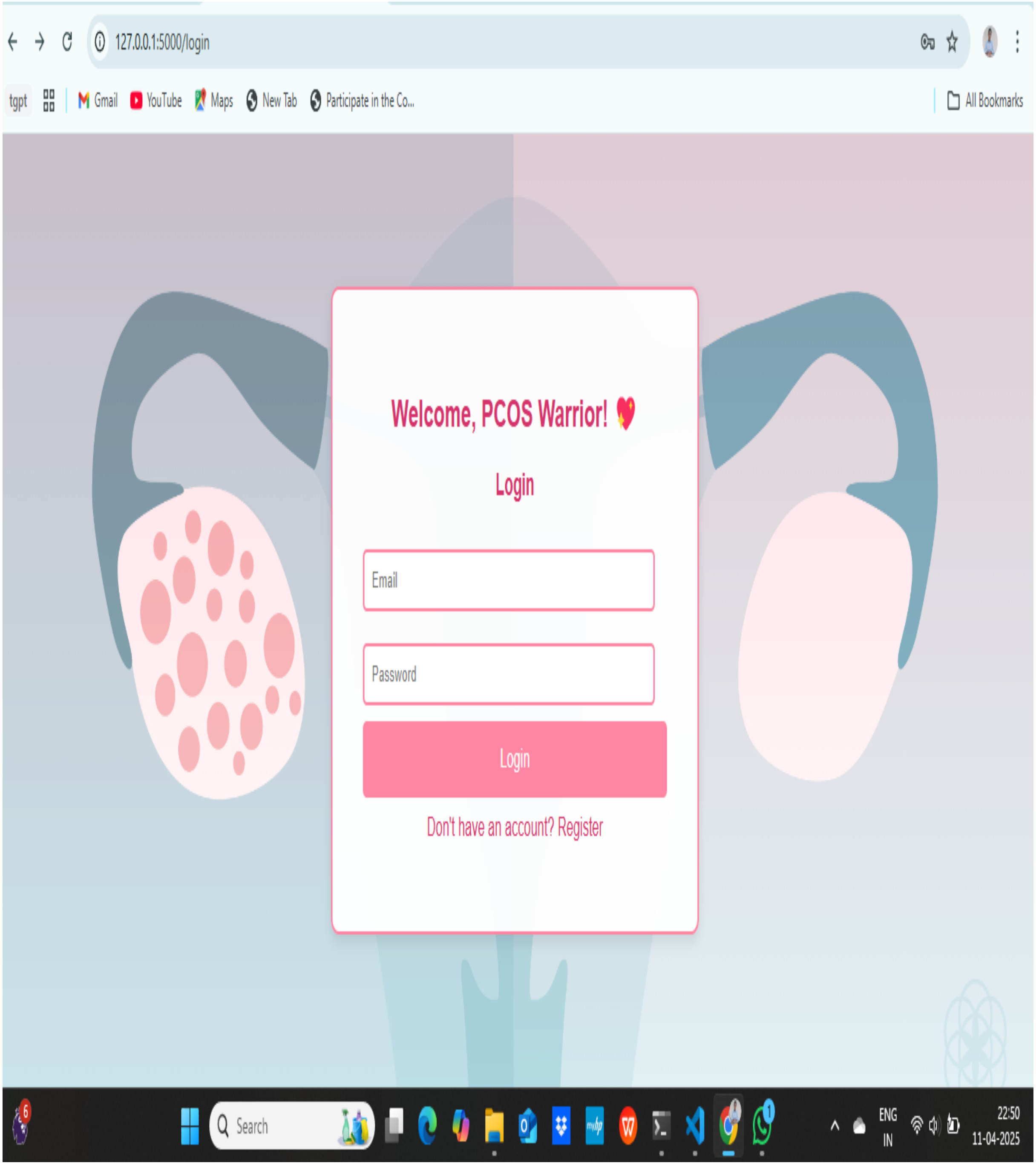


Figure 5.1: Login Page

- If a user does not already have an account, they can create one here to access personalized PCOS prediction and health analysis features.

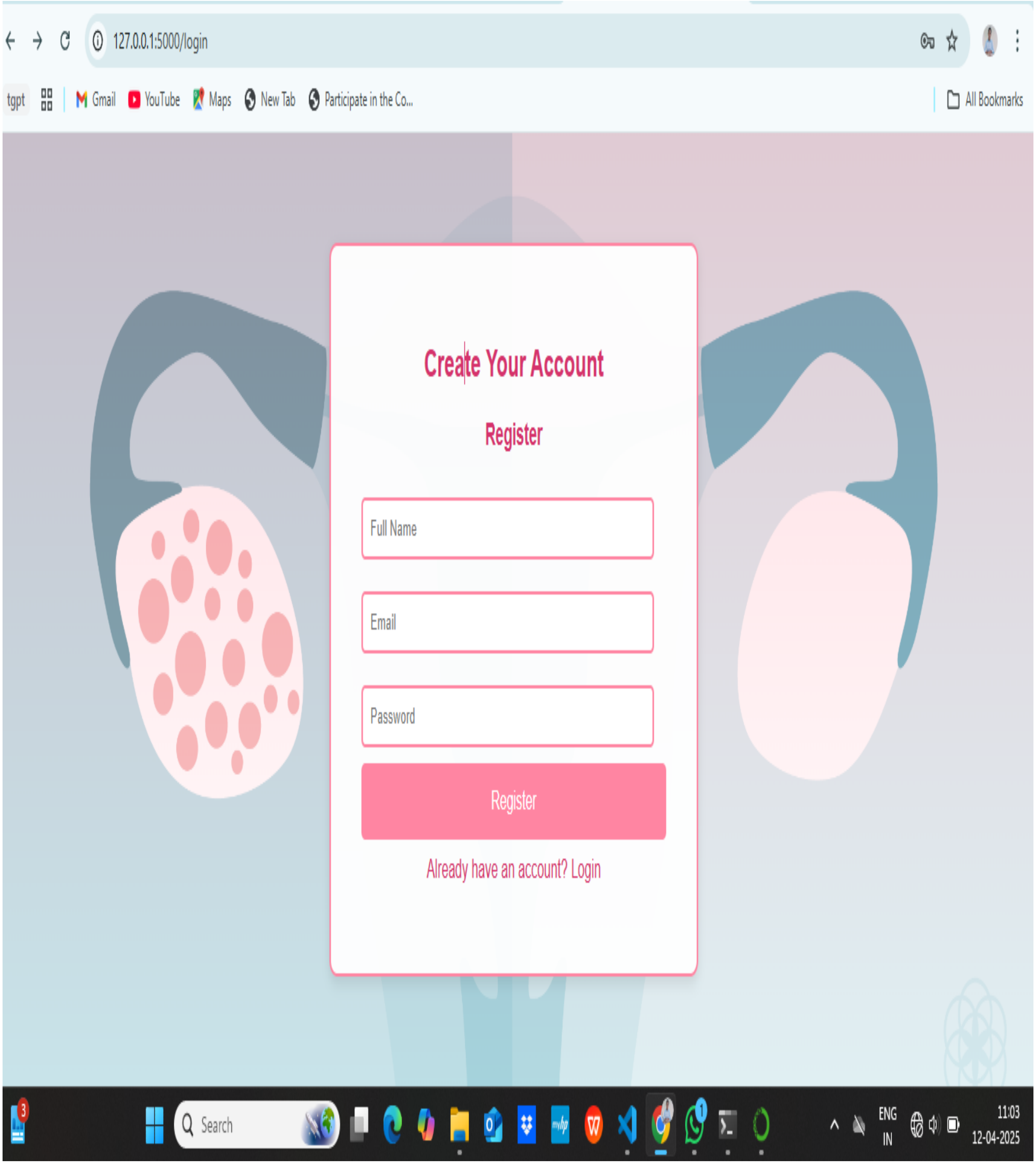


Figure 5.2: Registration Page

- This is the Home Page of our PCOS Detection App. The interface is designed to be simple and user-friendly for easy access to the app's core features.

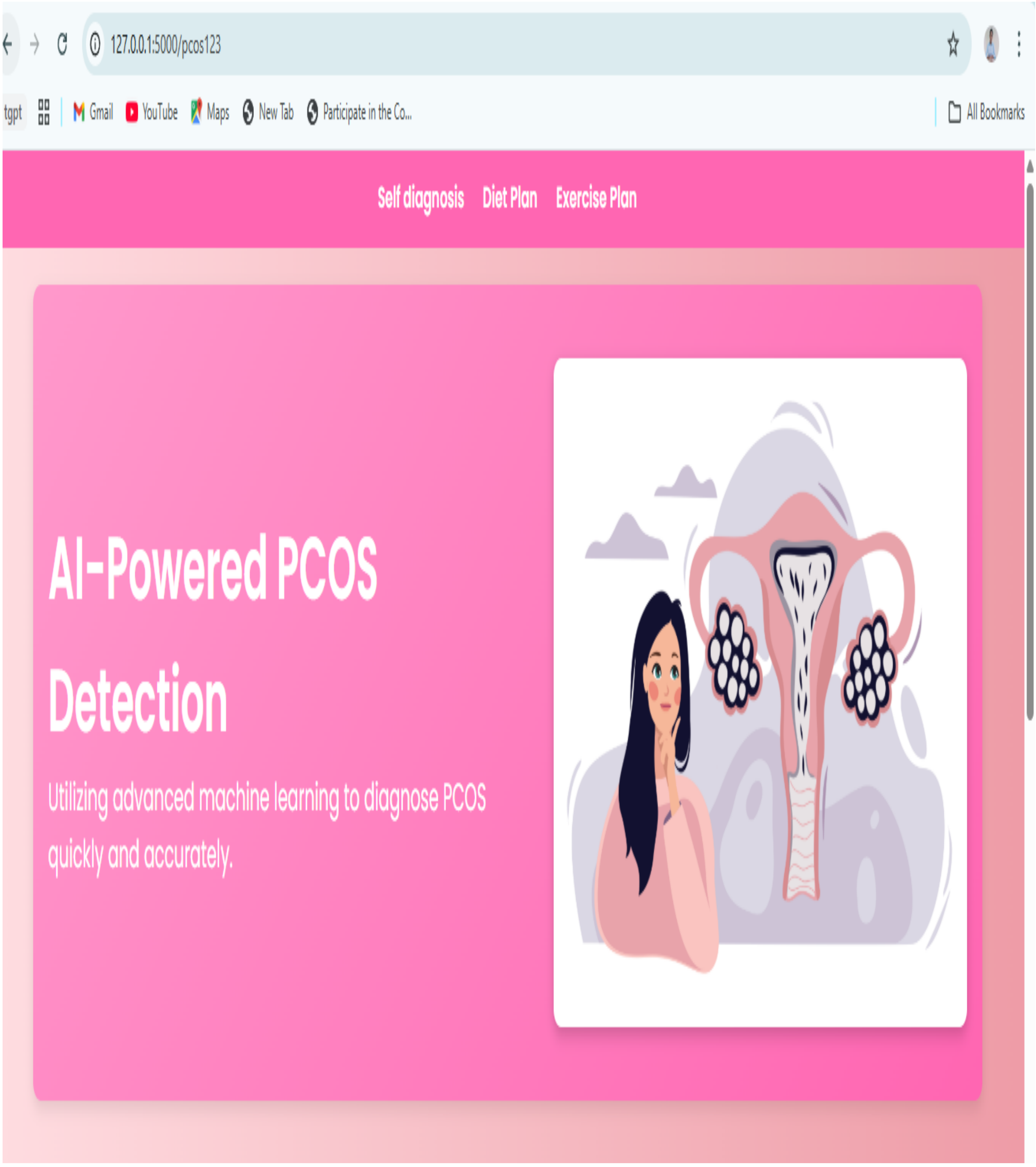


Figure 5.3: Home Page

- This is the User Input Page where individuals answer a series of health-related questions. These include details like age, weight, cycle length, and other symptoms relevant to PCOS.

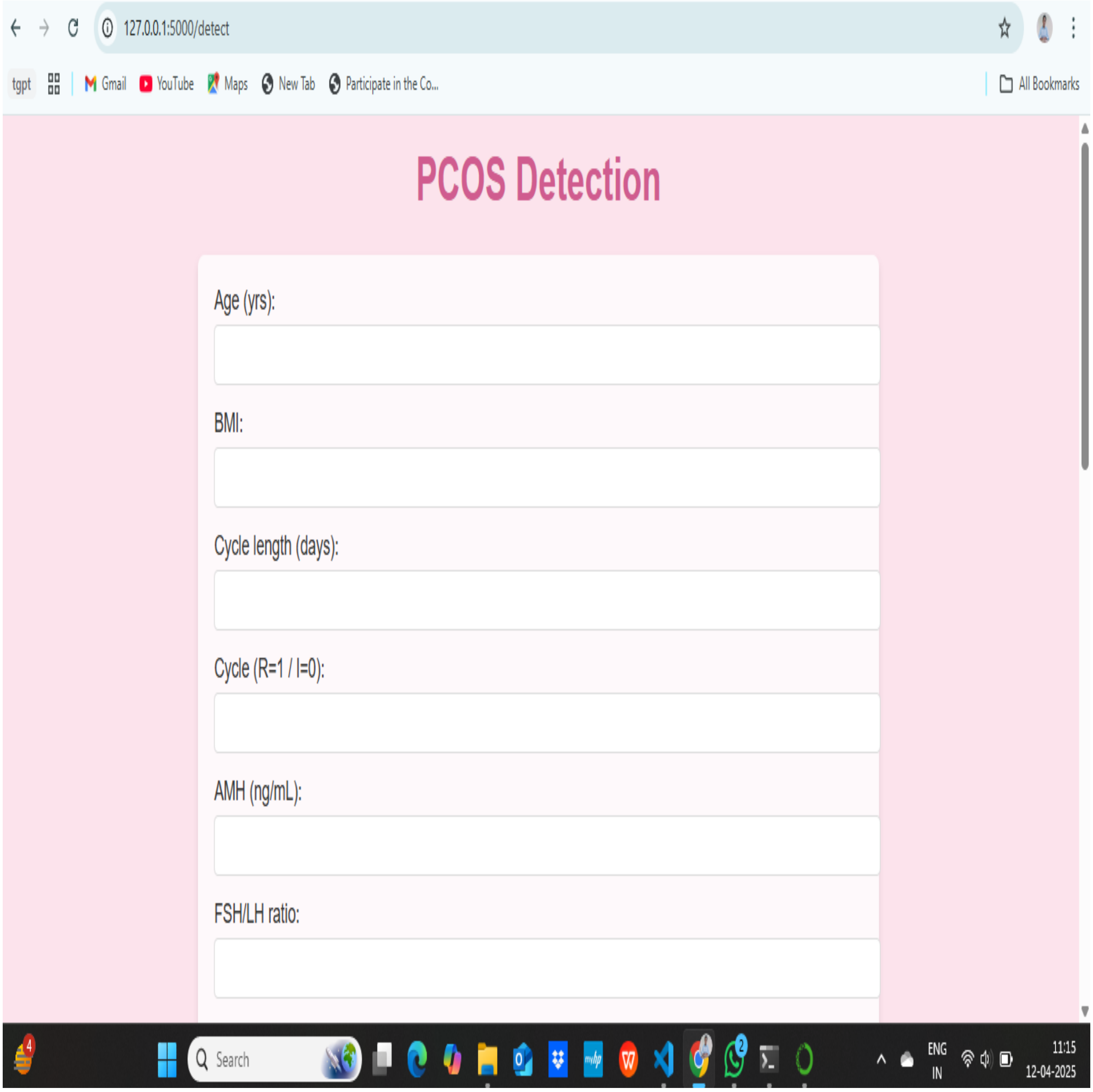


Figure 5.4: User Input Page

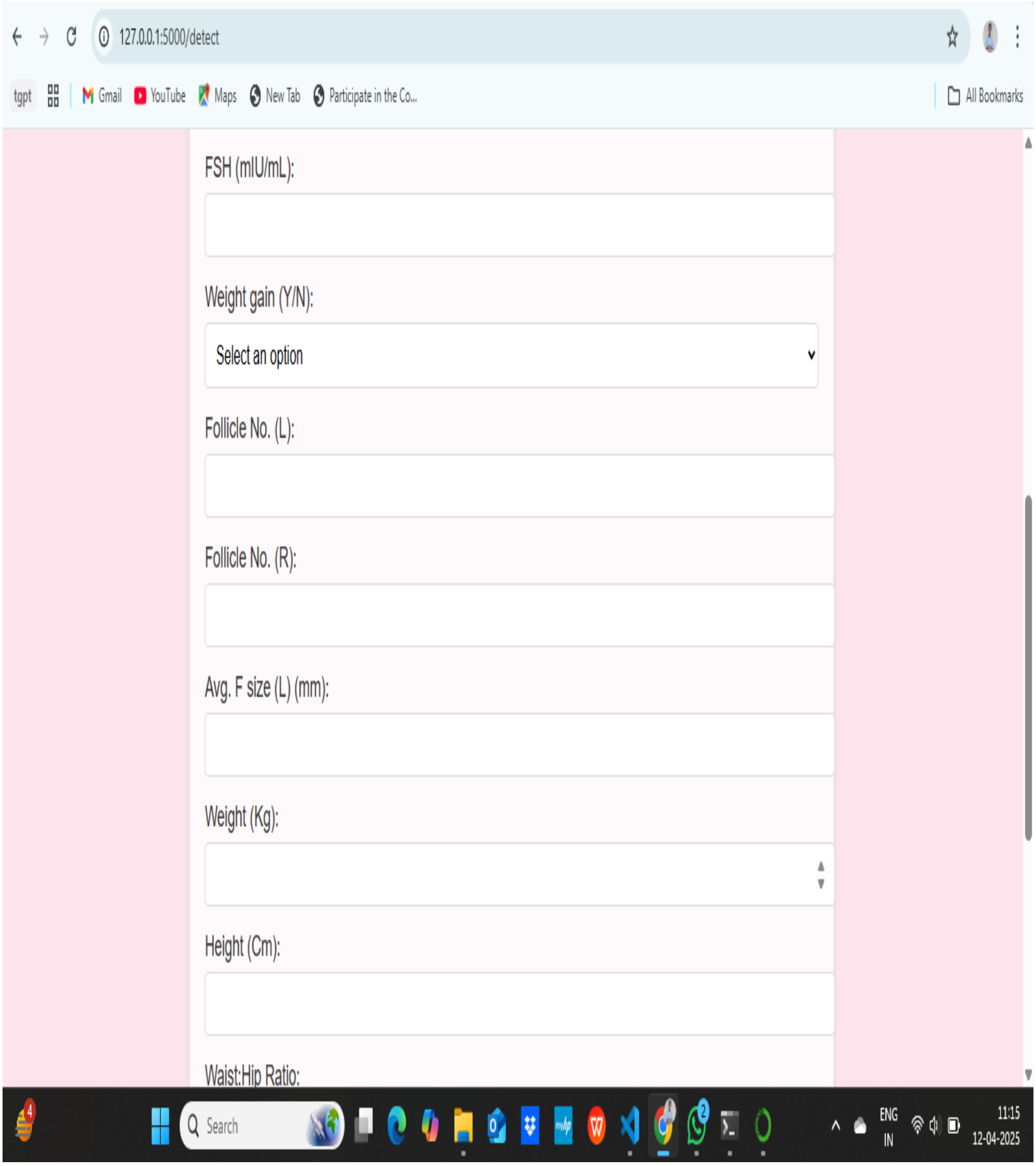


Figure 5.5: User Input Page

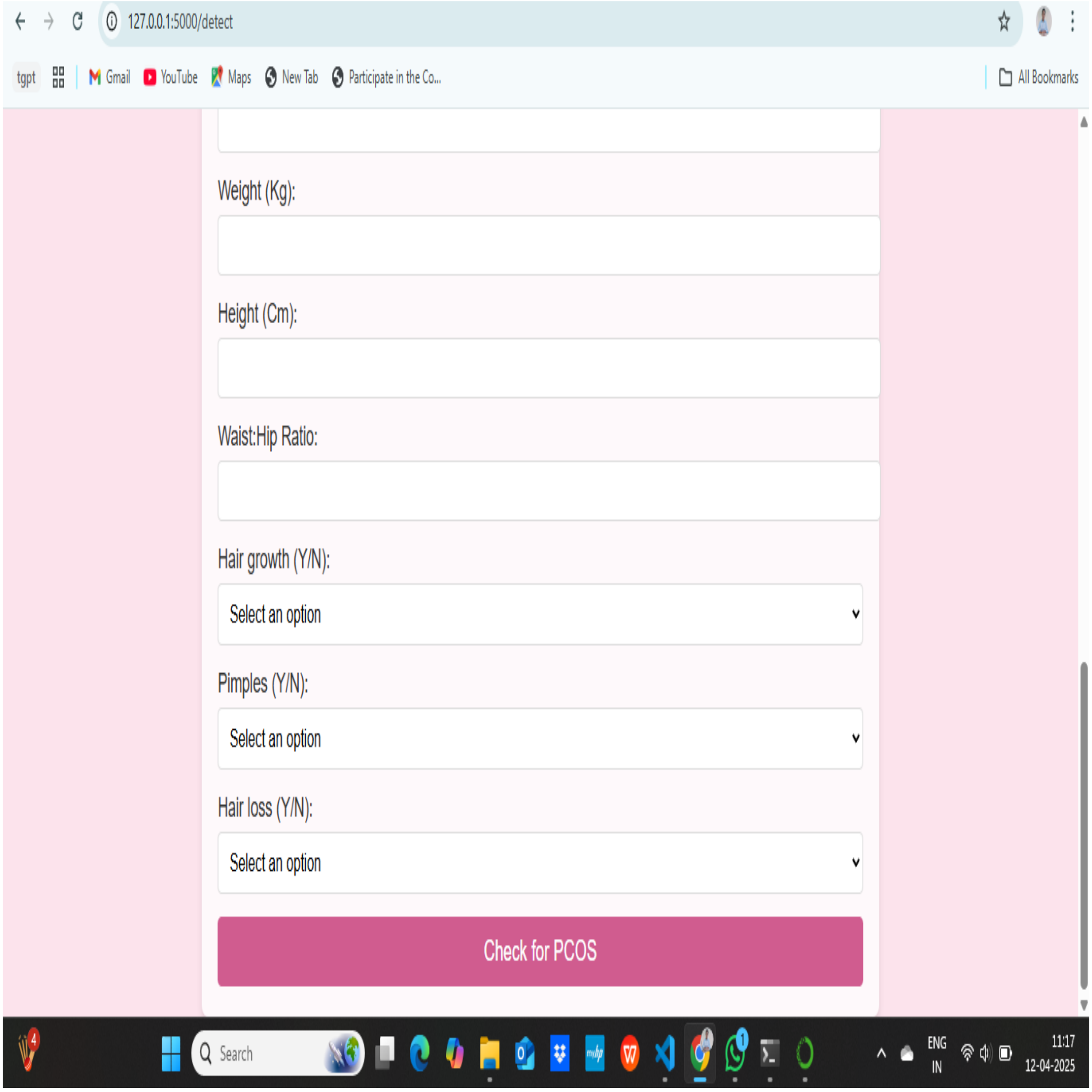
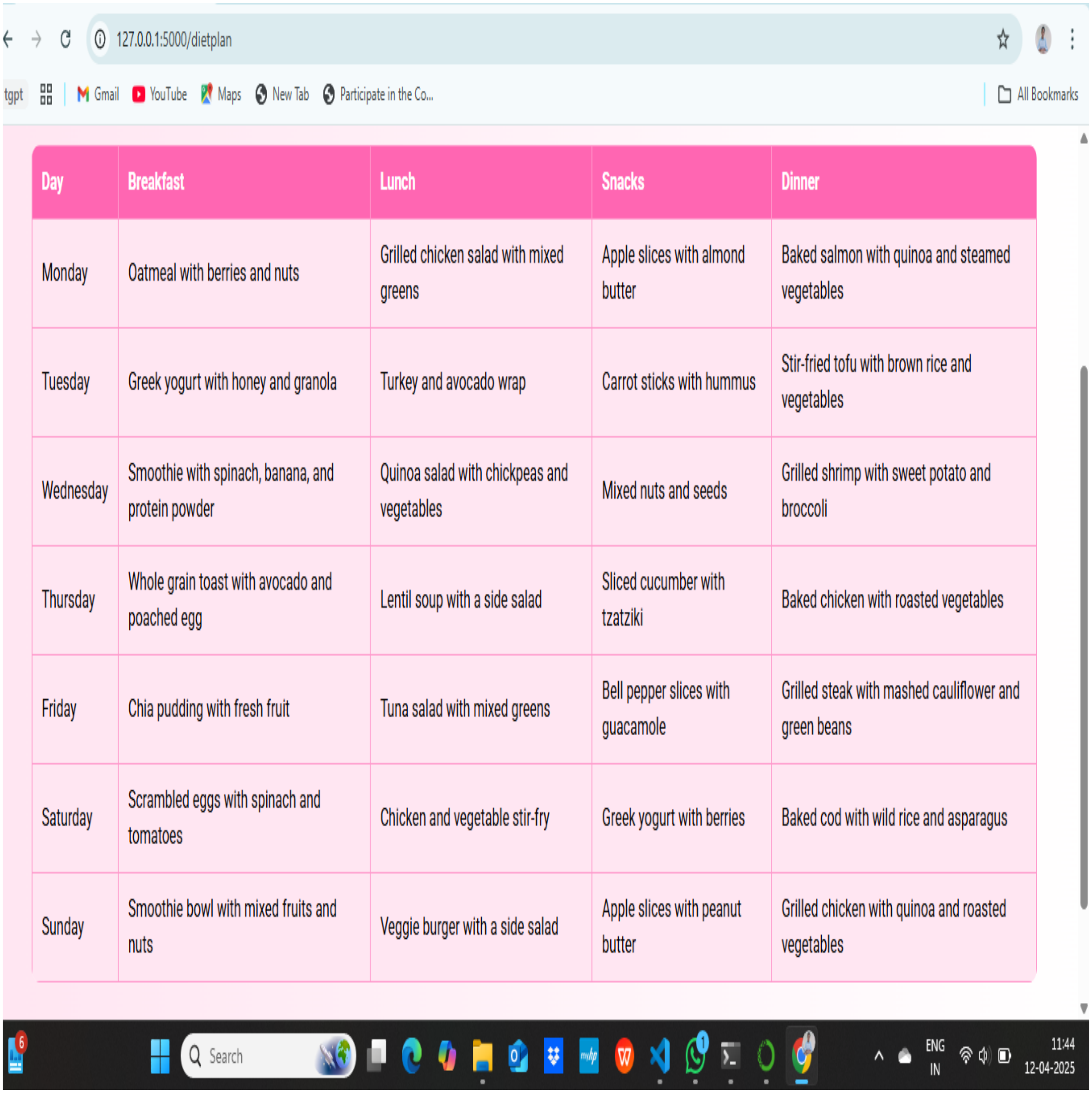


Figure 5.6: User Input Page

- A balanced diet plan for a person is crucial for maintaining overall health. Here's a diet plan that provides proper nutrients while also considering the specific needs of someone with PCOS



Day	Breakfast	Lunch	Snacks	Dinner
Monday	Oatmeal with berries and nuts	Grilled chicken salad with mixed greens	Apple slices with almond butter	Baked salmon with quinoa and steamed vegetables
Tuesday	Greek yogurt with honey and granola	Turkey and avocado wrap	Carrot sticks with hummus	Stir-fried tofu with brown rice and vegetables
Wednesday	Smoothie with spinach, banana, and protein powder	Quinoa salad with chickpeas and vegetables	Mixed nuts and seeds	Grilled shrimp with sweet potato and broccoli
Thursday	Whole grain toast with avocado and poached egg	Lentil soup with a side salad	Sliced cucumber with tzatziki	Baked chicken with roasted vegetables
Friday	Chia pudding with fresh fruit	Tuna salad with mixed greens	Bell pepper slices with guacamole	Grilled steak with mashed cauliflower and green beans
Saturday	Scrambled eggs with spinach and tomatoes	Chicken and vegetable stir-fry	Greek yogurt with berries	Baked cod with wild rice and asparagus
Sunday	Smoothie bowl with mixed fruits and nuts	Veggie burger with a side salad	Apple slices with peanut butter	Grilled chicken with quinoa and roasted vegetables

Figure 5.7: Diet Chart

- An exercise plan is essential for managing PCOS as it helps regulate insulin levels, maintain a healthy weight, and improve overall well-being. A combination of strength training, cardio, and flexibility exercises can be beneficial. Here's a balanced exercise plan for someone managing PCOS

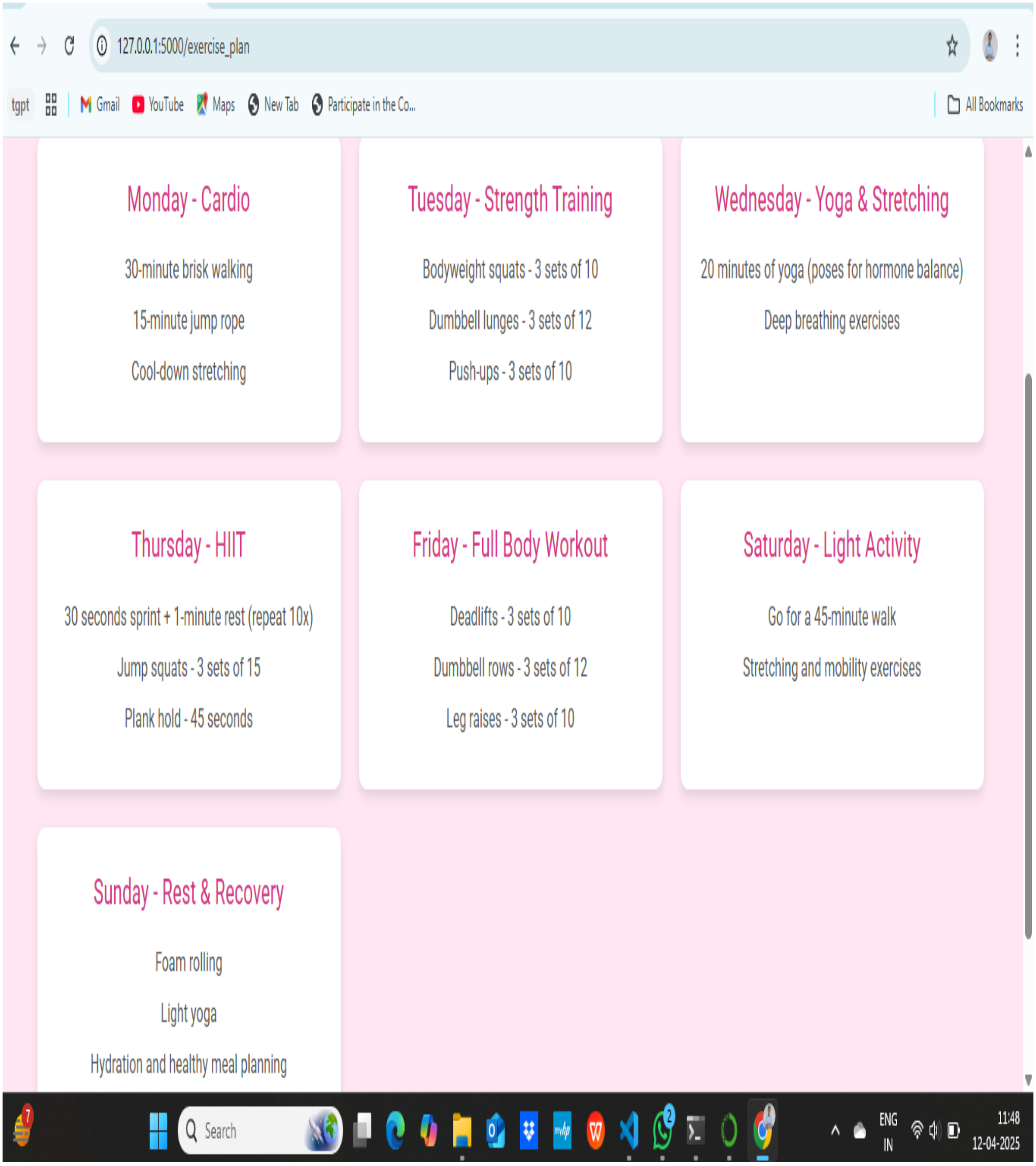


Figure 5.8: Exercise Chart

Chapter 6

Testing

In our project, "PCOS Detection Based on Machine Learning," we conducted three primary types of testing to ensure the accuracy, performance, and reliability of the application.
These testing types are:

- **Unit Testing:**
  Unit testing was carried out to verify individual components of the system such as specific functions, methods, or classes. This ensured that each logic unit, including data processing methods and machine learning functions, worked correctly and returned the expected output.

- **Integration Testing:**
  Integration testing was performed to test the interaction between different modules—such as data input, ML model prediction, and report generation. We used tools like emulators and mock data to verify that these modules functioned correctly when combined, and that the data flow between them remained consistent.

- **System Testing:**
  In system testing, the complete application was tested in a real-world scenario. The system was shared with users including healthcare professionals and test patients. Their feedback was gathered to assess usability, diagnosis accuracy, and overall experience. Based on the feedback, we made necessary improvements to better meet user require-

ments.

Test Case ID	Module / Function	Input	Expected Output	Actual Output
UT01 Validate Input Fields	Validate Input Fields	Age: 25, Weight: 6, Cycle L-nggth: 28	Input accepted	Pass
UT02 BMI Calculaton Function	BMI Calculatíon Function	Height: 160 cm, Weight: 6o kg	BMI = 23.4	Pass
UT03 ML Prediction Function	Valid health data (JSON format)	Output: PCOS Likelihood	Output: PCOS Likelihood = 80%	Pass
UT04 Report Generation	Diagnosis result = Positive	PDF report generated with PCOS	PDF report generated with PCOS positive indication	Pass
UT06 Invalid Input Handling	Age: „abc" Weight:–45	Error m displayed	Error message displayed	Pass

Figure 6.1: Unit Testing

Integration Testing Table

Test Case ID	Integration Scenario	Test Description	Expected Outcome	Actual Outcome	Status
IT01	Data Input and Authentication	Log in and input personal and health data	Data successfully submitted and stored	Data success fully submitted and stored	Pass
IT02	Data Submission and PCOS Prediction	Upload test results for PCOS prediction	Test results integrated and analyzed by system	Test results integrated and analyzed by system	Pass
IT03	Diagnosis Viewing	View PCOS diagnosis after prediction	PCOS diagnosis displayed	PCOS diagnosis displayed	Pass
IT04	Diagnosis Viewing $_{F1}$eport Generation	Generate diagnostic report based on diagnosis	Diagnostic report generated	Diagnostic report generated	Pass
IT05	Report Generation and Data Storage User $_5$ Provider Access	Save diagnostic report to system	Diagnostic report stored in 'Diagnosis Reports'	Diagnosis and report views accessible to patient and provider	Pass

Figure 6.2: Integration Testing

System Testing

Modules	Input	Expected Output	Result
PCOS Detection	Personal and health data	Diagnosis result should be generated based on the input data	Pass
PCOS Diagnosis	Diagnosis result	Displayed to the patient and healthcare provider	Pass
Provide Treatment Plan	Diagnosis result	Treatment plan should be offered based on diagnosis	Pass

Figure 6.3: System Testing

Chapter 7

Conclusion and Future Work

The project concluded with a successful implementation of a PCOS detection system using machine learning, overcoming challenges in data handling and model accuracy. It enhanced our practical skills and understanding of real-world health applications. Future work includes adding features like real-time health tracking, expert consultation, and improving model performance with larger datasets.

7.1 Conclusion

With the increasing health concerns among women, especially related to hormonal disorders, the need for early diagnosis and effective monitoring systems has become essential. PCOS (Polycystic Ovary Syndrome) is one of the most common health issues faced by women today, and due to lack of awareness and delayed diagnosis, many continue to suffer without proper treatment. To tackle this problem, a smart and accessible solution is necessary.

Our project, PCOS Detection Based on Machine Learning, aims to bridge this gap by providing an intelligent platform that helps in early prediction of PCOS using patient data. It assists users in understanding their condition at an early stage and encourages timely medical intervention. By combining technology with healthcare, the system supports both patients and healthcare professionals, contributing towards a more efficient and preventive approach to women's health.

7.2 Future Work

Although our project successfully implements machine learning models for the early detection of Polycystic Ovary Syndrome (PCOS), there is significant scope for enhancement and further research. Future improvements can be explored in the following directions:

- **Larger and More Diverse Dataset:** Our current model is trained on a limited dataset. Incorporating larger and more diverse patient data from multiple regions and age groups can improve the model's generalizability and performance across different populations.

- **Deep Learning Models:** Future work can include experimenting with deep learning techniques, such as neural networks, especially for complex feature relationships or unstructured data like medical images (e.g., ultrasound scans).

- **Real-Time Web or Mobile Application:** To enhance usability, the trained model can be integrated into a real-time web or mobile application, enabling healthcare providers and patients to use the system interactively for early diagnosis support.

- **Explainable AI (XAI):** Implementing explainability techniques such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) can help clinicians understand the reasoning behind model predictions, increasing trust and adoption.

- **Integration with Electronic Health Records (EHR):** Integrating the system with existing EHR platforms can streamline data input and provide continuous learning capabilities by updating the model with new patient data over time.

- **Multi-Disorder Detection System:** The system can be extended to detect other endocrine or reproductive disorders using similar approaches, transforming the project into a broader diagnostic platform.

- **Clinical Trials and Validation:** To move toward real-world deployment, collaboration with medical institutions for clinical validation and trials will be essential to evaluate the practical effectiveness and safety of the proposed solution.

Bibliography

- M. Sumathi, “Study and detection of PCOS related diseases using CNN,” *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1070, Nov. 2021, Art. no. 012062.

- Silva I, et al., “Polycystic ovary syndrome: clinical and laboratory variables related to new phenotypes using machine-learning models,” *J. Endocrinol. Invest.*, 2022, pp. 1–9.

- R.P.R. Murugan, “Application of Machine Learning in PCOS Diagnosis: A Survey,” published in 2021.

- P.K. Jain, “Predicting Polycystic Ovary Syndrome (PCOS) Using Machine Learning Algorithms,” published in 2019.

- K.M.S.R. Reddy, “Machine Learning Models for PCOS Diagnosis Using Clinical Data,” published in 2021.

- K.R. Anusha, “A Review on PCOS Diagnosis with Machine Learning Models,” published in 2020.

- Araviiskaia E., “The role of topical dermocosmetics in acne vulgaris,” *J. Eur. Acad. Dermatol. Venereol.*, vol. 30, pp. 926–935, 2016.

- M.H. Gold and D.J. Goldberg, “Current treatments of acne: Medications, lights, lasers, and a novel 650 µs 1064 nm Nd:YAG laser,” *J. Cosmet. Dermatol.*, vol. 16, pp. 303–318, 2017.

- H.K. Jaiswal, “Automated Detection of PCOS using Data Mining and Machine Learning,” published in 2020.

- Rohit Mishra et al., “A Predicate Based Fault Localization Technique Based on Test Case Reduction,” *International Journal of Scientific Technology Research*, vol. 4, no. 10, Oct. 2015.