A SYMPTOM TRACKING AND VISUALIZATION SYSTEM FOR WOMEN WITH POLYCYSTIC OVARY SYNDROME (PCOS) USING PYTHON

BY

ESOSA IFEOLUWA EMMANUEL-OSAGIE

PSC2105326

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF COMPUTING,

UNIVERSITY OF BENIN,

BENIN CITY,

EDO STATE, NIGERIA.

NOVEMBER 2025

A SYMPTOM TRACKING AND VISUALIZATION SYSTEM FOR WOMEN WITH POLYCYSTIC OVARY SYNDROME USING PYTHON

BY

ESOSA IFEOLUWA EMMANUEL-OSAGIE

PSC2105326

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF COMPUTING, UNIVERSITY OF BENIN, BENIN CITY

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF A BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE

NOVEMBER 2025

CERTIFICATION

This is to certify that this project work was carried out by **ESOSA IFEOLUWA EMMANUEL-OSAGIE** with Matriculation Number **PSC2105326** under my supervision. It is adequate and satisfactory, both in scope and content, for the award of Bachelor of Science (B.sc) Degree in Computer Science of the University of Benin.

PROF. (MRS) V.V.N AKWUKWUMA	**DATE**
Project Supervisor	

APPROVAL

This project work is hereby approved in partial fulfilment of the requirements for the award of Bachelor of Science (B.Sc.) Degree in Computer Science from the University of Benin.

PROF. (MRS) V.V.N AKWUKWUMA **DATE**

Project Supervisor

DR. (MRS.) R.A. USIOBAIFO **DATE**

Head of Department

DEDICATION

This project is dedicated to God Almighty for giving me the strength and wisdom to see it through to completion, and even throughout my stay in the University of Benin (UNIBEN). It is also dedicated to my parents; Mr and Mrs Osagie; for their love, support and guidance throughout my academic journey.

ACKNOWLEDGEMENT

My utmost acknowledgement goes to God Almighty for giving me the strength, wisdom and direction throughout my academic journey. I would like to express my gratitude to my project supervisor, Prof. (Mrs.) V.V.N. Akwukwuma for her consistent guidance towards ensuring the successful completion of this project.

I would also like to specially thank my project coordinator Dr. Maxwell Osagie, and other lecturers in the Department of Computer Science who I have been opportune to cross paths with, and have impacted me immensely these past few years: Prof. G.O. Ekuobase, Dr. (Mrs.) A.R. Usiobaifo, Prof. (Mrs.) A.O. Egwali, Dr. F.O. Oliha, Prof. K.C. Ukaoha, Prof. A.A. Imiavan, Prof. (Mrs.) F. Egbokhare, Prof. F.I. Amadin, Prof. (Mrs.) S. Konyeha, Prof. (Mrs.) V.I. Osubor, Dr. (Mrs.) Aziken, Dr. F.O. Chete, Dr. (Mrs) R.O. Osaseri, Dr. J.C. Obi, Mr. P. E.B. Imiefoh, Mr. I.E. Obasohan, Mr. S.O.P. Oliomogbe, Mr. K.O. Otokiti, Mr. I.E. obayagbonna, Mrs. R.I. Izevbizua, Mr. E.C. Igodan, Miss L.O.Usiosefe, Mr J. Okhuoya, Prof. F.A.U. Imouokhome, Mrs. J.I. Adun, Dr. E. Nweli and Dr. D.N. Idehen.

Finally, I also want to appreciate those who contributed to the success of this project: Ewarami-Yesin Courage, Ogbeide Faith Ebosetale, Ogamune Vera, Ofomola Faith Ufuoma, Osawaru Praise and Osayuwamen Favour. I would also like to thank my family and friends for their support, words of encouragement, and consistent guidance throughout this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Polycystic Ovary Syndrome (PCOS) is a chronic endocrine disorder requiring continuous monitoring to manage its complex and irregular symptoms. Current digital health applications often fail to address this need, providing generalized tracking, inadequate visualization, and insufficient privacy protection.

This project presents the design and development of a specialized, Python-based symptom tracking and visualization system to empower women with PCOS. Built using an Object-Oriented Analysis and Design (OOAD) methodology, the system ensures a modular and user-centered architecture. It leverages Matplotlib for generating insightful visualizations of symptoms, mood, and lifestyle factors. A Flask-based web interface offers an intuitive dashboard for data entry and review, with a commitment to user privacy through local data storage.

The resulting system demonstrates that open-source Python technologies can effectively bridge gaps in women's digital health by providing an affordable, transparent, and adaptable solution. It facilitates improved health literacy, data-driven self-management, and enhances patient-clinician communication, highlighting the significant potential of Python in advancing personalized chronic disease management.

CHAPTER ONE

INTRODUCTION

1.0 Background Study

Polycystic Ovary Syndrome (PCOS) is among the most common endocrine disorders that affect women in their reproductive age worldwide. According to Teede et al. (2018), epidemiologic studies in unselected populations in many countries have quoted a clinical prevalence of 5% to 15% among all reproductive-age women, though its global prevalence was estimated at 21.27% in recent studies by Deswal et al. (2020). It thus affects millions of women from different geographical regions, socio-economic backgrounds, and ethnic groups, hence has become an important international health concern.

PCOS is a complex endocrine disorder, basically characterized by excess production of androgen, irregular cycles of menstrual flow, and multiple small cysts on the ovaries. Common modes of presentation are inability to conceive or infertility, oligomenorrhoea or irregular periods, obesity, and hirsutism or excessive hair. Besides being related to reproductive health concerns, PCOS is associated with a number of comorbidities, which are insulin resistance, type 2 diabetes, cardiovascular disease, metabolic syndrome, depression, anxiety, and endometrial cancer.

Globally, the management of PCOS presents an important challenge. Poor knowledge about PCOS among both the general public and some medical professionals usually translates into late diagnosis and treatment. Furthermore, chronicity of the condition requires lifelong management, which is not only physically but also emotionally exhausting for the affected women. In addition, stigma associated with infertility and physical manifestations of PCOS may also hinder women from seeking medical care or openly discussing their condition.

At the moment, there is no cure for PCOS; management is aimed at alleviating symptoms and preventing long-term complications through lifestyle modification, medication, and close follow-up (Teede et al., 2018). The successful management of PCOS requires that women monitor several variables over time, which may include menstrual cycle patterns, body weight, symptoms such as acne and hirsutism, mood changes, dietary habits, physical activity level, and metabolic parameters. This again constitutes a very important part of the management that will establish symptom patterns, treatment efficacy, and complication prevention.

Traditional paper-based symptom tracking methods have many drawbacks. They may get easily lost or destroyed, are hardly kept correctly for a long time, and are cumbersome to analyze in search of patterns and trends. It is also exhausting to share them with healthcare providers. Additionally, manual tracking does not provide immediate feedback or visualization of symptom patterns that may make it hard for women to understand the relationship between different symptoms and triggers.

Digital health solutions have quickly emerged due to the rapid growth of technologies and an increase in the number of smartphone users across the world. This helps in the management of chronic diseases, as noted by Pong et al. (2024). The use of mHealth technologies, symptom-tracking applications, and data visualization tools presents new, hitherto unimaginable opportunities for improving PCOS management. These digital solutions provide for continuous monitoring, immediate feedback, pattern recognition, and activation of women in taking responsibility for their health.

However, the review of PCOS management applications has shown significant gaps: most applications deal with only one aspect of the condition; lack sophisticated visualization tools that can help users find symptom associations; and struggle for engagement over time. Besides, much concern about data privacy and security remains regarding cloud-based solutions, and users often hesitate to share sensitive health information.

Python is one of the leading programming languages for healthcare application development due to its flexibility, comprehensive libraries that allow for data analytics and visualization, and ease of deployment on multiple platforms. The integration of Python with SQLite allows the secure local storage of data, an important issue considering the concerns related to health data privacy.

The project proposes an integrated symptom tracking and visualization system, uniquely designed for women with PCOS; this solution extends Python's capabilities to overcome the limitations found in existing solutions. With this, women will be able to easily track multiple symptoms of PCOS over time; visually illustrate their data with interactive charts and graphs to identify trends and patterns; store health information securely locally; and generate detailed reports they can share with health professionals.

1.1 Research Motivation

The rationale for this study is the pressing need for accessible digital solutions to enhance the management of PCOS. Apart from the physical manifestation of symptoms, PCOS involves psychological and emotional issues such as depression, anxiety, and low self-esteem; these are usually aggravated by infertility and social stigmatization. Effective management requires continuous self-monitoring and the precise tracking of symptoms, which is hardly achievable with traditional methods.

The increased adoption of digital health tools, especially since the COVID-19 pandemic began, creates an opportunity to develop user-friendly systems that will improve communication between patients and their healthcare providers. Current apps are few in number and focused specifically on PCOS; most lack advanced tracking, data visualization, or protection of privacy.

It is, therefore, important that a Python symptom tracking and visualization system be designed to afford women with PCOS the opportunity to track their health, better communicate about treatment, and protect personal data.

1.2 Problem Statement

PCOS is a very common syndrome among females all over the world, and while symptom monitoring is important in managing the disease, several critical problems still remain. There's a lack of infrastructure for symptom monitoring because most women with PCOS rely on memory or simple paper-based tracking, which are unreliable, hard to adhere to, and lack analysis to discover patterns. This incomplete information often results in less optimal treatment outcomes. Fragmented healthcare experiences complicate matters, as women see various specialists; without a complete symptom and treatment record across all contacts, achieving continuity of care and determining effective interventions is challenging, requiring tools for comprehensive documentation and effective communication. There is also a severe lack of pattern recognition since PCOS symptoms are interconnected and influenced by many factors like diet, stress, and medication, making it hard for women to determine which factors worsen symptoms or which interventions work without visualization tools to uncover patterns and correlations for informed lifestyle changes. This leads to poor long-term management, as PCOS is a chronic condition requiring lifelong management, but regular tracking cannot be maintained over years without engaging, user-friendly tools; most current apps fail to continuously engage users, leading to incomplete data and missed opportunities for early intervention when symptom patterns change. Data privacy and security concerns are major barriers, as most commercially available health apps store sensitive user data on external servers, raising hesitation about tracking information like fertility or mental health unless solutions focus on local data storage and guarantee users full control over their personal health information. Cost barriers to access further limit management, as premium features in many commercial health apps require subscription fees that are unaffordable for many women, making access inequitable and necessitating the development of low-cost or no-cost comprehensive solutions. Furthermore, there is limited customization and flexibility in most available applications, which enforce rigid tracking templates that don't fit the diverse symptom patterns and management priorities of every woman with PCOS, reducing usefulness and adherence. Finally, even when data is tracked, inadequate visualization remains an issue, as most symptom-tracking applications offer data only as plain lists or simple charts that fail to reveal patterns or relationships between different variables; women need advanced, user-friendly visualization tools to transform tracked data into actionable insights. These issues ultimately result in the poor management of PCOS, increased complication risks, reduced quality of life among affected women, and unnecessary health costs, underscoring a great need for an inclusive, accessible, user-centered digital solution developed specifically to accommodate the complex needs of women living with PCOS.

1.3 Aim and Objectives

Aim: To develop a symptom tracking and visualization system in Python that is comprehensive but user-friendly for the effective monitoring and management of PCOS in women.

Objectives:

I. To develop a comprehensive symptom tracking system for PCOS-related signs.

ii. To visualize tracked data using Python’s libraries like Matplotlib.

iii. To provide secure local data storage for privacy.

iv. To design a simple, user-friendly interface.

v. To generate reports that support clinical decision-making.

1.4 Scope of Research

The project is aimed at the design and implementation of a symptom tracking and visualization system in Python, targeted specifically for women with PCOS. The system will have full symptom tracking, data visualization through interactive charts and graphs, secure local data storage via SQLite, and the ability to export for sharing with healthcare providers. It will have a user-friendly interface built with HTML, CSS and JavaScript; incorporate educational resources about the ways to manage PCOS; and include data backup capabilities to ensure information security and accessibility.

1.5 Significance of the Study

This Python-based symptom tracking and visualization system for women with PCOS has significant ramifications:

1. The system empowers women with PCOS to take control of their health by understanding symptom patterns and correlations.
2. It also promotes personalized healthcare reducing the risk of complications like diabetes and cardiovascular disease.
3. It also enhances doctor-patient communication for accurate diagnosis.
4. Finally, it ensures privacy since all data is stored locally.

1.6 Limitations of the Study

While this research project aims to develop a comprehensive symptom-tracking and visualization system for women with PCOS, several limitations define the boundaries of this study:

1. **Platform Limitation**: This system is built as a web application available via different devices using a browser. It is not a native mobile application for either iOS or Android platforms, so some of the functionalities associated with specific devices, such as push notifications or offline capability, may be supported to a limited extent.
2. **Device Integration**: The system does not integrate with wearable devices or other health tracking devices for automated data collection. Users themselves have to input all the symptom data manually, which may affect consistency and adherence.
3. **Advanced Analytics**: Artificial intelligence or machine learning algorithms for predictive analytics and personalized recommendations are beyond the scope of this project. The system developed is focused on data visualization and pattern recognition based on traditional methods of analytics.
4. **Healthcare System Integration**: The product is not directly integrated with electronic health records or hospital information systems. Data will be shared with healthcare providers via exported reports, not through seamless system-to-system integration.
5. **Communication Features**: The system does not include telemedicine features or direct consultation with healthcare providers, nor does it include social networking or community forums to provide support for patients with PCOS.
6. **Clinical Validation**: This research does not involve comprehensive clinical validation studies or randomized controlled trials to rigorously evaluate the system's impact on clinical outcomes. The evaluation is limited to usability testing with potential users.
7. **Language Support**: The initial version of the system is developed in the English language only. Multi-language support to serve diverse global populations is not included in the current scope.
8. **Sample Size**: Testing and evaluation are conducted with a limited number of users through the deployed web application and thus may not fully represent the diverse experiences and needs of all women with PCOS from different demographics and geographical locations.

CHAPTER TWO

LITERATURE REVIEW

This chapter introduces the evolving role that technology plays in women's health, focusing on a condition seldom discussed: Polycystic Ovary Syndrome. It foreshadows the rising importance of symptom tracking, not just for diagnosis but to understand one's personal health patterns over time. By examining existing systems for symptom tracking in PCOS, the methods of data collection, and the technological basis behind such systems, this chapter gradually reveals why it may be necessary to go in a different direction with something simpler. This review sets a basis for a design that combines medical insight with accessible, everyday technology.

2.1 Overview Of PCOS

Polycystic Ovary Syndrome (PCOS) is a common endocrine disorder that affects women of reproductive age, with prevalence estimates ranging from 6% to 20% worldwide depending on diagnostic criteria (Azziz et al., 2016). It is characterized by a combination of symptoms that may include irregular menstrual cycles, excess androgen levels, and polycystic ovarian morphology. The condition is multifactorial in nature, with its exact cause still under investigation, though it is widely associated with hormonal imbalance, insulin resistance, and genetic predisposition.

The chronicity and heterogeneity of PCOS require an integrated management approach beyond simple symptomatic relief. This, in turn, requires a very attentive, continuous surveillance of symptoms to capture the dynamic nature of the condition. Systematic recording of key health metrics, such as menstrual cycle dysfunction, mood swings, fluctuations in body weight, changes in skin, and any other relevant measures, needs to be logged. Such longitudinal data collection can form the basis for insights into the disease course that will enable clinicians to provide truly personalized treatment.

Recent progress in digital health technologies has made tools available that enable structured symptom monitoring, with both mobile applications and desktop-based platforms becoming increasingly available for logging and analyzing data related to the condition. However, many of the existing solutions have limitations, such as requiring internet connectivity, complicated functionalities that could be beyond the technical proficiency of most users, or embedding their functionality in generalized menstrual trackers that are ill-equipped to address the specific needs of women diagnosed with PCOS.

In this regard, the proposed project aims at the creation of an easy-to-use, user-friendly digital system specifically for tracking and visualizing symptoms in PCOS and capable of working offline. The core philosophy of simplicity and ease of access will be prioritized in the presented system, with intuitive graphical representation of health data, to empower women suffering from PCOS to understand patterns in their health. This can facilitate informed and shared decision-making with healthcare professionals for better disease management.

2.1.1 Key Features of PCOS

PCOS is diagnosed by the Rotterdam criteria of 2003, which require at least two of the following:
1. **Ovulatory Dysfunction**: Irregular or absent menstrual cycles, such as oligomenorrhea (fewer than nine periods annually) or amenorrhea (Rotterdam ESHRE/ASRM-Sponsored PCOS Consensus Workshop Group, 2004).
2. **Hyperandrogenism**: The presence of excess androgen, which may be evident as hirsutism, acne, or androgenic alopecia (Fauser et al., 2012).
3. **Polycystic Ovaries**: Several ovarian cysts during an ultrasound test, although not all women who are diagnosed have cysts present (Boyle & Teede, 2016).

2.1.2 Symptoms and Long-term Risks

The symptoms and risks associated with PCOS are varied and include:

• **Reproductiv**e: Irregular menstruation, infertility, and recurrent miscarriages. (Teede et al., 2018)

• **Metabolic**: Weight gain is associated with insulin resistance; this in turn increases the risk of getting type 2 diabetes and cardiovascular diseases. (Moran et al., 2010).

• **Dermatological**: Acne, hirsute, and tags on the skin with oily skin also. (Goodarzi et al., 2011).

• **Psychological**: Anxiety and depression linked with challenging physical health and a lower quality of life. (Dokras et al., 2018).

• **Others**: Depression and fatigue, as well as pelvic pain and sleep changes. (Sirmans & Pate, 2014).

2.1.3 Why PCOS Needs to Be Monitored

The World Health Organization (WHO) points out that conditions such as PCOS need long-term health monitoring since modification of signs in the early stages can prevent further complications. The reasons why monitoring is particularly essential in this case are as follows:

1. **Early Detection of Complications**: PCOS is associated with long-term risks like diabetes, cardiovascular diseases, and even endometrial cancer. Women diagnosed with PCOS appear to have at least a 2-4 times higher risk of diabetes (WHO, 2023). Monitoring metabolic and some reproductive signs, including weight, glucose, and even amenorrhea, helps in early complication detection. This becomes particularly essential in the reduction of morbidity.

2. **Fertility Management**: According to the NIH in 2022, 70-80% of PCOS-related infertility can be managed in due time with ovulation monitoring. Menstrual and hormonal signs, such as basal body temperature tracking, can help not only in identifying the fertile window but also in guiding some fertility treatments, like ovulation induction with clomiphene.

3. **Psychological Health Monitoring**: PCOS affects the psychological condition of women through anxiety and depression and is considered one of the major causes of reduced quality of life; indeed 40% of the women affected by PCOS develop these psychiatric disorders (Dokras et al., 2018). Teede et al., (2021) in their meta-analysis on JAMA Psychiatry noted that women with the disease had 3 times greater rates of depression, much of it associated with the severity of symptoms beyond a certain threshold.

2.2 Existing Symptom Tracking And Management Systems For Women With PCOS

2.2.1 Traditional Methods

Before the advent of mHealth applications and electronic record-keeping, symptom monitoring for PCOS relied mainly on manual, paper-based means. These include symptom recordings in diaries, notebooks, printed calendars, or physician-provided symptom log sheets.

1. **Paper-based symptom diaries or journals**

In the traditional approach to managing PCOS, many women keep diaries or symptom trackers in some form of printed charts or templates, recording relevant symptoms daily or weekly. The date, alongside the period dates, its duration and flow, and the hirsutism, acne, weight changes, and mood swings, are recorded. This will also record contextual entries that include diet, physical activity, and levels of stress that can affect symptoms. The symptoms are measured in terms of severity and recorded on a scale of 1 to 10. These records are made and reviewed periodically, allowing the patient and healthcare provider to notice trends; for instance, irregular cycles or the worsening of symptoms over time. According to Teede et al. (2018), menstrual irregularity and symptom severity tracking allows refinement of treatment approaches, making them more proactive and tailored for the patient by the inclusion of, for example, oral contraceptives and changes in lifestyle.

2. **Menstrual Cycle Calendars**

A calendar-based methodology is widely employed for the monitoring of menstrual cycles, a core feature of Polycystic Ovary Syndrome (PCOS) associated with ovulatory dysfunction. Utilizing a calendar-based approach, individuals record the onset and cessation of menstrual periods, document characteristics of menstrual flow, and note potential indicators of ovulation, such as alterations in cervical mucus. This may be accomplished using physical calendars or printed templates. Additionally, some individuals incorporate measurements of basal body temperature or results from ovulation predictor kits. Longitudinal tracking of these parameters facilitates the identification of cycle duration, irregularities such as amenorrhea, and patterns indicative of oligomenorrhea. For instance, a record might document, “Menstruation: October 1–5, light flow” followed by “No menstruation in November.” As noted by the World Health Organization (WHO, 2023), irregular menstrual cycles are a defining characteristic of PCOS, and systematic tracking is instrumental in supporting fertility planning and diagnosing ovulatory disorders.

3. **Symptom checklists or tables**

The use of structured checklists or tabular formats represents an effective approach for systematically documenting multiple symptoms associated with Polycystic Ovary Syndrome (PCOS). These tools, which may be developed by healthcare professionals or designed by individuals, provide a framework for recording symptoms such as acne, hirsutism, weight fluctuations, and mood variations alongside corresponding dates. Users can indicate the presence of symptoms, quantify their severity using a numerical scale, or include detailed observations. For instance, a table might feature columns labelled “Date,” “Menstruation (Yes/No),” “Acne (Scale 1–10),” “Mood (Scale 1–10),” and “Additional Notes.” This structured approach facilitates the identification of correlations between symptoms, such as the exacerbation of hirsutism associated with weight gain, which is a critical consideration in managing insulin resistance (Moran et al., 2010).

2.2.2 Limitations of Traditional Manual Methods

Traditional paper-based symptom-tracking techniques for PCOS are very straightforward, accessible, and easy to use; however, they have many limitations that make them less effective for long-term condition management.

1. **Time-consuming**: Manual logging implies frequent entries, usually daily or weekly, and analysis of data, such as average menstrual cycle length or symptom frequency evaluation, done manually. This is a highly labor-intensive practice that will mostly result in human error, especially when dealing with more than one symptom for a length of time.

2 **Lack of Data Visualization**: Manual methods, unlike digital platforms, cannot provide the option to automatically create visual outputs such as charts, graphs, or trend analyses. This in turn will impede the patient's and healthcare professional's ability to identify patterns quickly, such as the relationship between stress and menstrual disturbances.

3 **Inconsistent Data Entry:** Maintaining a very accurate and consistent long-term record requires a great amount of discipline. Any omission or irregular logging reduces the accuracy of the data, hence limiting meaningful interpretations that could be made from the records.

4 **Challenges in Data Storage and Sharing**: Due to their physical nature, paper-based records are prone to loss, physical damage, or deterioration with age. Further, sharing of such records with healthcare professionals is often done through physical transport or manual transcription, which delays clinical decision-making.

5 **Limited Analytical Functionality**: Manual tracking tools, by design, have only limited functionalities for in-depth analyses, such as statistical evaluations of relationships between weight fluctuations and ovulatory patterns. In contrast, computational approaches like those developed with Python can efficiently carry out such an analysis, allowing the detection of critical symptom interlinkages that might otherwise remain unrecognized, thus constraining optimization of treatment strategies.

2.2.3 Mobile Health Applications in PCOS Monitoring

A. Mainstream Period Trackers

1. **Flo**: Flo is a popular application among women to track menstrual cycles, ovulation, pregnancy, and even a broad array of symptoms; it also encompasses tools designed specifically for Polycystic Ovary Syndrome (PCOS) management. Its 'Symptom Checker' runs the entered data to find patterns and possible risks while being able to trace more than 70 symptoms, such as irregular periods, acne, weight fluctuations, mood changes, etc. AI-driven algorithms predict cycles with an accuracy of up to 95% and suggest tailored lifestyle adjustments.

The enhancements added in 2025 gave Flo an extended analytical capability that allowed the detection of symptom patterns with much more precision, including links to insulin resistance-a big factor in PCOS. The app visualizes trends with interactive charts, symptom timelines, and heatmaps. Meanwhile, integration with popular wearables like Fitbit and Apple Watch adds extra health metrics for the user. Premium analytics and virtual health assistant tools offer greater functionality; however, some functions require a subscription.

In one large-scale study, Jain et al. (2021) analyzed the user data of Flo from 142 countries and established bloating, menstrual irregularities, and higher BMI as strong predictors of PCOS, thus confirming the app's potential as a valuable mHealth tool for symptom monitoring and characterization in diverse populations. User feedback is overwhelmingly positive, with over seven million 5-star reviews praising its accuracy, educational resources, and supportive community features.

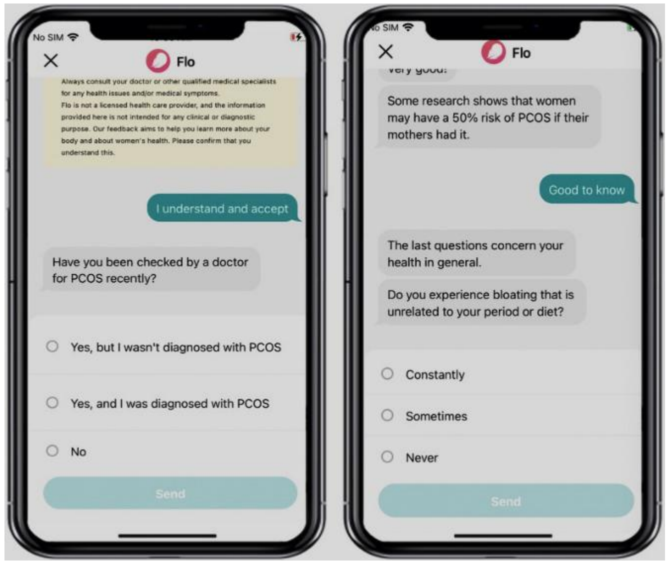

Figure 2.1: Display of the Flo app

2. **Clue**: Clue is a science-based menstrual and ovulation tracking application designed to arm users with data-driven insights into their cycles. It is particularly helpful for women suffering from Polycystic Ovary Syndrome, as the app can adapt to irregular cycles and track an extensive range of symptoms related to menstrual flow, pain, mood, energy, and sleep. Its predictive algorithms adjust to variation and offer accurate forecasts of menstruation, ovulation, and PMS symptoms. In users with PCOS, Clue brings in long-term pattern analysis to identify potential hormonal imbalances, supplemented by a PCOS risk assessment tool that reviews symptoms like irregular cycles and androgen-related changes (Clue, 2025). A study conducted by Rodriguez et al. in 2020 has shown that the tracking of cycle irregularities and symptoms in apps like Clue can help in early identification and thus is clinically important.

Along with tracking, Clue offers a variety of educational content on PCOS and associated complications: depression, anxiety, and metabolic issues. Updates in 2025 continue to improve its predictive analytics for cycles and symptom trend analysis, enhancing the app's utility in clinical discussions. Data presentation features include interactive cycle calendars, hormone trend charts, and symptom progression graphs. Integrations with Apple Health and Google Fit allow for imports of both activity and sleep data.

Clue was founded in 2013 by Ida Tin in Berlin, launched with a €50,000 budget, and has since grown to millions of active users worldwide, particularly in the United States. The company secured €20 million in funding in November 2024 to expand its mHealth services (Tin, 2024). However, it does not yet provide in-depth metabolic tracking such as insulin monitoring, and PCOS-specific community features remain limited.

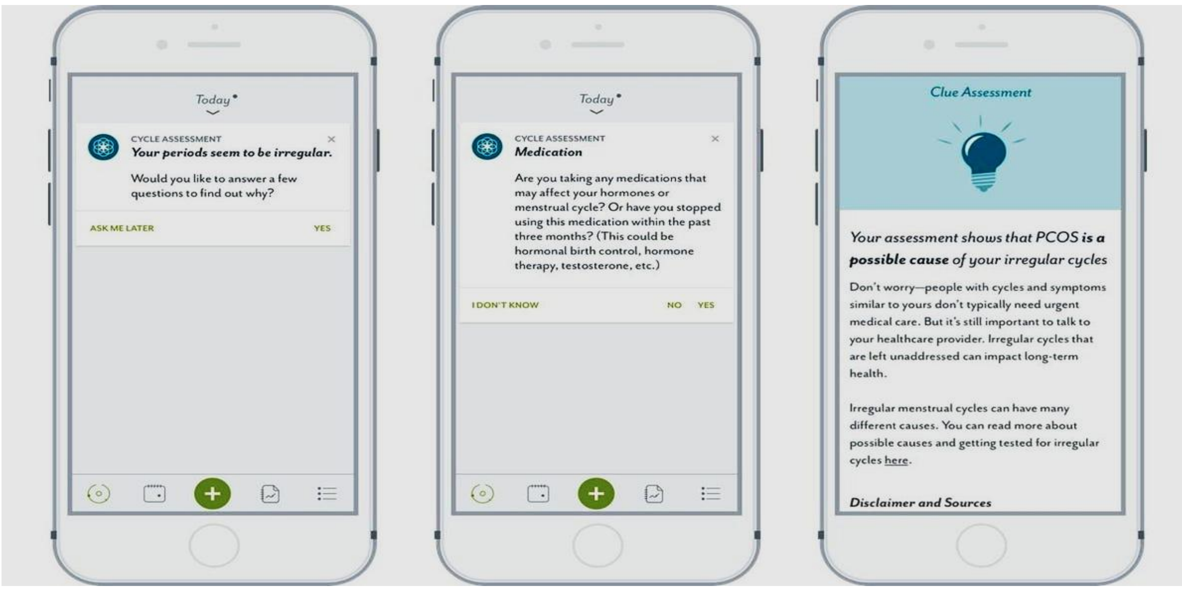

Figure 2.2: Display of the Cue app

3. **MyFLO**: MyFLO, developed by Alisa Vitti, the founder of Flo Living, is an application in which one syncs with their women's cycles, particularly for PCOS management. This app tracks day-to-day symptoms like cramps, mood swings, energy levels, and cravings while offering phase-specific suggestions for diet, exercise, and self-care during the follicular, ovulatory, luteal, and menstrual phases. This will help alleviate any hormonal imbalances that a woman with PCOS may face, such as estrogen dominance versus progesterone deficiency, as stated by Vitti (2020).

The app offers educational resources about PCOS, focusing on nutrition-based strategies to ameliorate symptoms. By 2025, MyFLO had grown to support individuals from menarche through perimenopause, providing irregular cycle tracking and personalized "biohacks" to optimize hormonal health. Visualization tools include phase calendars, symptom charts, and progress graphs that highlight hormonal changes and improvements over time.

While there is limited integration, MyFLO does interface with calendar applications for phase-based planning and is able to integrate with basic health trackers. Predictive analytics leverage logged data to predict symptom patterns and offer optimal timing for activities, even for users with irregular PCOS cycles.

While praised for its innovative cycle-syncing methodology, it is designed to be more user-friendly than medically diagnostic-supportive-when compared to other platforms like Flo-and accuracy is contingent on regular logging of symptoms therein. (Rodriguez et al., 2020)

App	**PCOS-Specific Features**	**Gaps**	**Citation**
Flo	"PCOS Mode" for irregular cycles	No insulin/glucose logging	*Flo Health White Paper*, 2023
Clue	Custom tags for symptoms	Limited metabolic tracking	*NPJ Digital Medicine*, 2022
MyFLO	Hormone-focused advice	$99/year paywall for advanced analytics	*Journal of Endocrine Society*, 2021

Table 2.1: Comparison of PCOS Features and Gaps in Selected Health Apps

B. **PCOS Specialized Apps**

These target PCOS directly, but suffer from poor scalability and validation.

1. **PCOS Diva**:

The PCOS Diva app was created by MB360, LLC, which was founded by a certified health coach and PCOS management author named Amy Medling. The app delivers a 14-day program aimed at reducing sugar cravings, leveling out blood sugar, and balancing hormones through meal plans, recipes, and lifestyle reminders. It helps users track goals and monitor symptoms like fatigue, mood swings, and inflammation. Educational resources are also included, based on Medling's book Healing PCOS: A 21-Day Plan for Reclaiming Your Health and Life with Polycystic Ovary Syndrome by Medling (2018). Launched in 2023, it provides a structured nutrition and lifestyle approach to managing PCOS. The app also connects with Apple Health and Fitbit to track sleep and activity, showing simple progress through checklists and some basic charts. Advanced features would include comprehensive symptom tracking or detailed visualization of long-term health patterns; these are not part of the app. In 2025, user reviews indicated usability issues and unmet expectations for broader PCOS management. Although the strength of PCOS Diva is its widely accessible nutrition-based program, the limitations found highlight the necessity for tools that enable detailed symptom tracking over time and visualization of patterns. This is where this particular project attempts to fill in these lacunas by providing structured logging with clear graphical outputs for long-term health management. (Solat et al. ,2025)

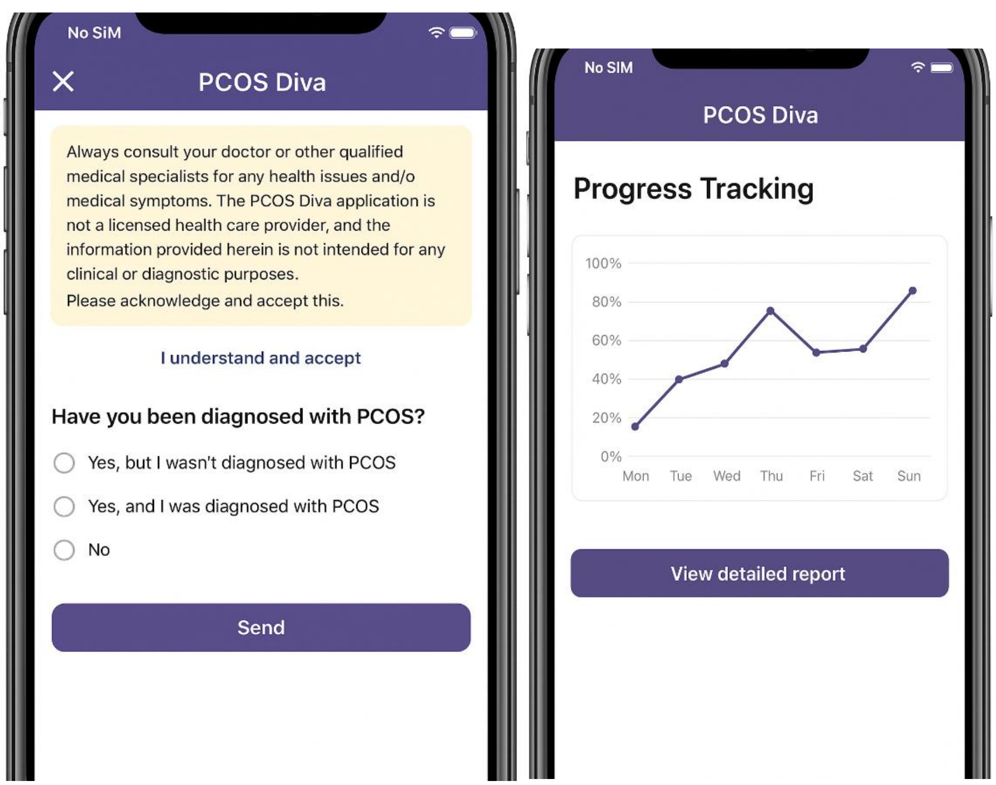

Figure 2.3: Display of the PCOS Diva app

2. **Ovia Health**:

Ovia Health by Ovuline, Inc. includes a range of applications, such as the Ovia Cycle & Pregnancy Tracker, that help track women's health through various stages of their lives, from menstruation to fertility, pregnancy, and menopause. For PCOS users, this app allows customized tracking of more than 100 symptoms-such as moods, nutrition, and sexual activity-in addition to recording birth control administration, providing support for irregular cycles, and offering users access to more than 2,000 expert articles and community forums.

The app includes interactive visualizations such as cycle calendars, symptom timelines, and pregnancy progress trackers. It integrates with Apple Health and Fitbit for importing steps, sleep, and weight, so users can correlate lifestyle data with PCOS-related health indicators. Predictive algorithms provide fertility scores, ovulation forecasts, and personalized recovery guidance while adapting to the irregular cycle.

Key features are stage-specific tracking, predictive modeling that is accurate, and resources and support free of charge, while anonymized data contributes to the advancement of women's health research as well (Consumer Reports, 2020; Ovia Health, 2023). Predictive models are prone to overfitting, assuming that all PCOS users are TTC and may very well neglect broader PCOS issues, including long-term metabolic risks. Many studies have shown that PCOS is associated with sleep and stress disruptions with malfunctions of the hypothalamic-pituitary-adrenal axis. Ovia does not provide obvious connections between these symptoms and their possible associations (Klose et al., 2015; Kahal et al., 2017; Caughey et al., 2022). Commonly shared concerns involve low usage, variable degrees of information quality, privacy concerns, and reduced accuracy for highly irregular cycles without premium access (Arabkermani et al., 2025; Consumer Reports, 2020).

Table 2.2: Comparative Table of PCOS-Relevant Features in Digital Health Tools

Feature	**Flo**	**Clue**	**MyFLO**	**PCOS Diva**	**Ovia**
Irregular	✓	✓	✗	✗	✓

cycle algorithm					
Symptom correlation tools	✗	✗	✗	✗	✗
Open-source backend	✗	✗	✗	✗	✗
HIPAA compliance	✗	✗	✗	✗	✓

2.2.4 Wearable Devices and IoT in Women's Health

The rapid adoption of consumer wearables and IoT devices has created new opportunities for continuous, passive health monitoring beyond episodic clinic visits. For a complex condition like PCOS, which affects reproductive, metabolic, and psychological health, these tools provide valuable real-world data that can support personalized management.

A. Categories of Relevant Wearables and IoT Devices

1. Wearable consumer wellness devices like Fitbit, Apple Watch, Garmin, Oura Ring, and Whoop capture heart rate, heart rate variability, sleep stages, activity level, and skin temperature. These metrics are particularly relevant for PCOS, given that sleep disturbances and resting tachycardia are common signs of stress and metabolic burden. By tracking physical activity, this further helps in managing weight, an important factor for PCOS comorbidities (Shin et al., 2021).

2. Continuous glucose monitors, like Dexcom G6/G7 and FreeStyle Libre, add another dimension by minutely recording glucose values to provide a full 24-hour glycemic profile. This is important, as up to 70% of women with PCOS have insulin resistance. Such data allows the person to understand how food choices, stress, or sleep may be impacting their blood sugar levels and make changes in response.

3. Devices for tracking fertility and basal body temperature, such as Tempdrop, Ava bracelet, and OvuSense, use subtle changes in body temperature or respiration that may be undetected by women but are supposed to signal ovulation. It is very helpful, especially for women with PCOS, when cycles are irregular and ovulation cannot be detected normally. Such devices automate temperature tracking and eliminate most of the untrustworthiness of traditional manual methods (Shalom et al., 2022).

2.3 Data Visualization in Healthcare

2.3.1 The Crucial Role of Visualization in Medicine

The brain is wired to process visual information far quicker than text or numbers, and visualization, therefore, is an essential tool in modern healthcare. Symptoms and variables for chronic conditions such as PCOS must be monitored over time, and this requires an understanding that is greatly enhanced by the use of visualization.

One major benefit is pattern recognition; it is hard for both patients and clinicians to detect meaningful trends in raw data, such as spreadsheets or written logs. The visualization tools transform the data into intuitive graphs and timelines, showing symptom cycles, possible triggers, and responses to treatment.

Another advantage lies in patient-provider communication; a simple chart or graph during limited clinic visits can almost serve as a common language, enabling easier discussions about health progress and changes in treatment compared with relying on memory or subjective descriptions. (Tisminetzky et al., 2022)

Lastly, the visualization encourages adherence as well as self-efficacy. When improvements are visual and measurable-for example, a downward trend in glucose levels following dietary changes-the patient is rewarded by positive behavior, reinforcing their sense of control in managing PCOS.

Visualization Technique	**Best For**	**PCOS Application Example**
Line Plot	Showing trends over time. The workhorse of temporal data.	Plotting weight, resting heart rate, or average glucose levels over a 3-month period.
Bar Chart	Comparing categorical data or discrete quantities.	Comparing average sleep duration or stress levels per week or month.
Scatter Plot	Identifying correlations and relationships between two variables.	Revealing a potential relationship between sleep quality (x-axis) and acne severity (y-axis).
Heatmap	Visualizing the intensity of a phenomenon across two dimensions (e.g., time and symptom).	Displaying a monthly calendar view where each cell's color indicates pain level, energy, or mood.
Correlation Matrix	Summarizing the strength of relationships between multiple variables at once.	A single graphic showing how weight, sleep, stress, and glucose all interrelate.

Table 2.3: Common Data Visualization Techniques and Their Application in PCOS Health Monitoring.

2.4 The Rise of Python in Medical Research and Applications

Python has become the leading programming language in data science, bioinformatics, and computational biology because of simplicity, flexibility, and a strong ecosystem of libraries. Its usage in healthcare has expanded from academia to clinical decision support systems and patient-focused applications.

It is mainly being adopted due to its readability and the capability it offers for rapid prototyping. Its simple syntax makes it intuitive, allowing clinicians, researchers, and developers to quickly turn ideas into working prototypes-an essential strength in rapid digital health innovation.

The scientific computing stack is another driver. Core libraries like NumPy for numerical computing, Pandas for data manipulation, Matplotlib and Seaborn for visualization, and Scikit-learn for machine learning form a free, integrated toolset that is competitive for the high-priced proprietary systems SAS and MATLAB (McKinney, 2022).

2.4.1 Key Application Areas Related to PCOS

Python's flexibility makes it suitable across different areas of healthcare, but in the context of PCOS management, certain applications prove more relevant:

1. Medical Data Analysis and Clinical Decision Support:
Clinical data analysis and predictive model construction are major roles played by such Python libraries as Pandas and Scikit-learn. For example, Zhou et al. (2023) developed an explainable machine learning framework using these tools, which achieved more than 92% accuracy in predicting PCOS based on clinical traits.

2. Medical Imaging:
Although it is not the central point of this project, Python is also used for medical imaging, supported by libraries such as OpenCV, Scikit-image, TensorFlow, and PyTorch. These tools are applied in automating pelvic ultrasound analysis by measuring follicle counts and ovarian volume, which are critical diagnostic indicators of PCOS.

2.4.2 The Case for a Python-Based PCOS Symptom Tracker

This project places Python in the position of being an effective foundation for a patient-centered PCOS symptom-tracking system. The selection of Matplotlib for visualization is based on several key advantages that it offers:

• **Exploratory Flexibility**: Unlike commercial applications that use fixed dashboards, a Python-based system can allow users to do ad hoc analyses and test personal hypotheses directly on their data.
• **Affordability and Accessibility**: Python and its ecosystem are completely free and open-source, eliminating subscription barriers and aiding in the democratization of digital health tools.
• **Data Ownership and Privacy:** All computations happen locally on the user's device, where sensitive health data remains under personal control. This addresses directly the privacy concerns associated with applications requiring computation in the cloud.
• **Scalable Foundation for Future Features**: The Pandas and Matplotlib build a flexible base upon which one can expand.
• **Integration**: Data from wearables like Fitbit or Oura can be imported to pandas DataFrames using APIs.
• **Predictive Modeling**: Once structured, these data can be extended to machine learning workflows using Scikit-learn to develop personalized alerts-for example, forecasting cycle irregularities based on stress, sleep, and lifestyle patterns.

Beyond the technical capabilities outlined above, the methodological foundation for this study is also supported by established software engineering principles that have been successfully applied to human-centered health systems. Akwukwuma, Obi, and Atamewan (2013) implemented a software-engineering–based model to mitigate Repetitive Strain Injury (RSI) among computer users, demonstrating that structured engineering frameworks can be adapted to address health-related problems. In later work, Akwukwuma (2024) explored user interface optimization through animation and visual design, underscoring the significance of usability and engagement; key factors for effective health-tracking platforms. The application of software engineering techniques provides a conceptual and methodological foundation for the present study, which adapts similar principles to a Python-based system for symptom tracking and visualization.

2.5 Gaps in the Existing Literature and Justification for the Study

While there has been considerable advancement in the mHealth technologies for women, there is still an important gap regarding design tools that are appropriate for the multi-dimensional nature of PCOS. A review of literature and existing digital applications reveals some key shortcomings:

1. **Lack of PCOS-Specific Functionality**

The most common health applications, including Flo, Clue, and Ovia, were designed initially either as general menstrual or fertility trackers. As their designs are still optimized for regular cycles and fertility goals, not all of them have updated features for PCOS, though some of them have added PCOS features. Thus, they often fall short when users are experiencing extreme cycle irregularity or when their main concerns go beyond conception (Rodriguez et al., 2020). The data captured by these platforms usually does not cover all the manifestations of PCOS, such as hirsutism, acne, insulin resistance, and mood disorders. For this reason, incomplete data collection restricts the generation of personalized insights and holistic management plans (Arabkermani et al., 2025).

2. **Barriers of Cost and Data Privacy**

The market is dominated by commercial apps operating on proprietary platforms, often locking advanced analytics and specialized features behind paywalls. This subscription model makes it prohibitively expensive, especially in low- and middle-income countries with very high PCOS prevalence and scarce healthcare resources (Teede et al., 2018). Added to these issues are serious data privacy concerns; indeed, investigations have found that certain apps share sensitive user data with third parties without adequate transparency or consent, undermining their trust for long-term health management (Consumer Reports, 2020).

3. **Poor Data Synthesis and Visualization**

Although digital apps represent an improvement over error-prone manual tracking methods, most of them present data in rudimentary formats, such as lists or static calendars. In the case of a chronic condition like PCOS, symptom-lifestyle-treatment interrelationships can only be uncovered with sophisticated, interactive visualizations of long-term trends. It is confirmed by research that such tools are crucial for both improving patient self-management and clinical decision-making, yet they are noticeably missing from the current offerings.

4. **Narrow Research Focus**

In comparison with other areas, such as pregnancy or contraception, PCOS remains disproportionately underrepresented in digital health research. Many existing studies also strongly focus on fertility outcomes, with there being a lack of evidence-based tools that support the long-term, day-to-day management of symptoms relating to PCOS (Jain et al., 2024). This is frustrating and causes users to disengage when forced to use generic apps not tailored for their needs (Arabkermani et al., 2025).

2.5.1 Rationale for the Proposed Study

This study is motivated by the critical lack of any dedicated, accessible, and evidence-based digital tool for PCOS management. For this reason, the proposed symptom tracking and visualization system in Python will directly address the documented shortcomings of the existing solutions through several key contributions:

1. **Holistic and Condition-Specific Tracking**

While commercially available apps focus on fertility, this system is architected to capture the complex nature of PCOS. The approach enables the integrated tracking of reproductive, metabolic, and psychological symptoms-from cycle irregularity and hirsutism to changes in mood. This comprehensive approach will be critical in delineating complex interrelationships between symptoms and tailoring appropriate, personalized effective management strategies where the capability to do so by mainstream applications is lacking (Teede et al., 2018).

2. **Accessible and Transparent Technological Framework**:
The proposed system is implemented using open-source Python libraries, such as Matplotlib for visualization, to ensure completely free access with full transparency. For the ease of a non-technical user, a lightweight web interface is built using the Flask framework. This allows for a user-friendly portal to interact with the core analysis features of the system, and is thus an easy translation from a research prototype to a tool for practical healthcare application without losing flexibility or control. (McKinney, 2022)

3. **Advanced Interactive Visualization of Data**

It offers a range of dynamic and customizable visualizations. These serve to assist users in active exploration of long-term trends and correlations, such as those relating to dietary habits and sleep quality with symptom severity. Such enhanced visual capabilities are highly important in building the self-efficacy of patients, ensuring better clinician-patient dialogues, and supporting adherence to complex lifestyle interventions.

4. **Bridging a Critical Gap between Research and Practice**
This project responds to a major gap in the market and academia: from exclusively fertility-oriented tracking to holistic, long-term condition management. Taking this open, transparent, and customizable platform will not only be instrumental for practical use by patients but also establish a validated foundation for future clinical research in developing evidence-based digital health interventions. The proof comes from Zhou et al. (2023).

2.6 Related Works

Author(s)	Year	Objectives	Major Features
Zahra Arabkermani, Saheed Barzegari, Atefeh Rouhani,	2025	To assess the content and quality of mobile apps	Symptom and activity tracking, community engagement, relaxation

Nahavandi N, Gibreel O, Arpaci I		developed for patients with PCOS using the Mobile App Rating Scale (MARS).	techniques, education and skill development, data sharing capabilities, password protection.
Ritu Deswal, Vinay Narwal, Amita Dang, Chandra S Pundir	2020	Systematic review of PCOS prevalence	Identified pooled mean prevalence of 21.27%; highlighted diagnostic challenges.
Candelyn Pong, Rachel Marjorie Wei Wen Tseng, Yih Chung Tham, Elaine Lum	2024	Review of digital health implementation in chronic disease management	Identified implementation frameworks and strategies; highlighted barriers and enablers.
Shengqing Zhou, Jiang Wu, Shuangxi Wang, Yang Zhang, Tianyuan Feng	2024	Development of digital health platform for chronic disease management in China	Demonstrated improved effectiveness through digital active health management.
Tarun Jain, Olivia Negris, Dannielle Brown, Isabel Galic, Rodion Salimgaraev, Liudmila Zhaunova	2021	To analyze PCOS symptoms using a menstrual tracking app (Flo)	Analyzed data from 142 countries; identified bloating as most frequent symptom; revealed symptoms broader than Rotterdam criteria.
Pamela J Wright, Charlotte Burts, Carolyn Harmon,	2025	To identify digital technology available and used	Mobile apps for menstrual/symptom tracking, education,

Cynthia F Corbett		by women with PCOS for PCOS-specific knowledge, self-management, or social support through a scoping review	lifestyle modification, and social support; wearables for real-time data collection; machine learning for predictive accuracy; SMS for weight loss promotion; web-based platforms weight management.
Haneul Lee, Seon-Heui Lee	2023	To evaluate the effectiveness of an integrated mobile application for lifestyle modifications in overweight women with PCOS.	Management of PCOS related symptoms and symptom scores, communication with researchers for counseling; feedback via messages or calls; goal setting for exercise and caloric intake
Anisha Sasha, Aporna Roy, Barsha Chakraborty, Bidishina Saha, Dipwanita Chowdhury	2023	To predict PCOS by comparing the efficacy of various machine learning algorithms.	Implements and compares Random Forest, and SVM models; uses feature selection to identify key clinical risk factors such as BMI

Table 2.4: Related Works on Polycystic Ovary Syndrome (PCOS) and Digital Health Interventions.

CHAPTER THREE

METHODOLOGY

3.1 System Analysis

System analysis is a fundamental phase in software development that involves examining and understanding the requirements, processes, and components necessary to create an effective solution. For this system, this phase focused on identifying the specific challenges women with PCOS face in managing their condition and determining how technology can address these challenges.

Several methodologies exist for system analysis and design, with the two primary approaches being:

1. **Object-Oriented Analysis and Design (OOAD):** This approach models the system as a collection of interacting objects, each with defined attributes and behaviors. OOAD emphasizes encapsulation, inheritance, and polymorphism, making it ideal for systems requiring modularity and scalability.

2. **Structured Systems Analysis and Design (SSAD):** This methodology focuses on decomposing the system into functional processes and data flows, using tools like Data Flow Diagrams and structured charts.

For this PCOS symptom tracking system, the Object-Oriented Analysis and Design (OOAD) methodology was selected because it aligns perfectly with the system's requirements. The system naturally maps to real-world entities such as users, symptom records, menstrual cycles, and visualizations, which can be effectively modeled as objects with specific attributes and methods. OOAD's emphasis on creating reusable, modular components ensure that the system can be easily maintained and extended in the future, for instance, adding new symptom types or visualization options without requiring major architectural changes.

3.2 Analysis of Existing System

Currently, women with PCOS rely predominantly on traditional paper-based methods for tracking their symptoms. This typically involves maintaining physical journals or notebooks where they manually record menstrual cycles, weight changes, symptoms, mood variations, and other health-related information. Some women use simple spreadsheet applications or general-purpose note-taking apps on their computers or smartphones, though these are not specifically designed for health tracking or PCOS management.

Healthcare providers often request that patients maintain symptom diaries, but these records are usually incomplete, inconsistent, or difficult to interpret during consultations. The existing approach lacks structure, making it challenging for both patients and providers to identify patterns or correlations between different symptoms and lifestyle factors.

Several commercial period tracking and general health apps exist in the market, such as Flo, Clue, and MyFlo. While these applications offer some relevant features, they are primarily designed for general menstrual cycle tracking or fitness monitoring rather than comprehensive PCOS management. They typically lack the sophisticated data visualization capabilities needed to reveal complex symptom relationships, do not accommodate the irregular cycles common in PCOS, and often store data on external servers, raising privacy concerns.

Additionally, these commercial apps usually require subscription fees for advanced features, creating financial barriers for many users. The educational content provided is often generic and not specifically tailored to PCOS management needs.

3.3 Problems of Existing System

The analysis of existing symptom tracking methods reveals several critical problems:

1. **Inconsistent Data Recording**: Paper-based tracking is prone to inconsistency. Women often forget to record symptoms regularly, resulting in incomplete data that limits the ability to identify meaningful patterns or evaluate treatment effectiveness.

2. **Lack of Pattern Recognition Capability**: Manual methods provide no automatic analysis or visualization of recorded data. Identifying relationships between symptoms, triggers, and lifestyle factors requires time-consuming manual review and interpretation.

3. **Poor Data Portability**: Physical journals cannot be easily shared with healthcare providers. Even when shared, providers must spend consultation time reviewing handwritten notes rather than quickly analyzing organized, visualized data.

4. **Limited Customization**: Commercial apps typically use standardized tracking parameters that may not capture the unique symptom combinations relevant to individual PCOS patients.

5. **Privacy and Security Concerns**: Cloud-based health apps store sensitive personal health information on external servers, raising concerns about data privacy, security breaches, and unauthorized access or use of personal health data.

6. **Cost Barriers**: Premium features in commercial apps often require ongoing subscription fees, making comprehensive symptom tracking tools inaccessible to women with limited financial resources.

7. **No Integration of Educational Resources**: Existing solutions rarely combine symptom tracking with evidence-based educational content specifically relevant to PCOS management.

3.4 Overview of Proposed System

The proposed system is a comprehensive Python-based symptom tracking and visualization application specifically designed to address the unique needs of women with PCOS. The system leverages modern web technologies and Python's powerful data analysis capabilities to provide an integrated solution for symptom monitoring, data visualization, and health management support.

For this system to completely achieve its goals, it must be able to perform the following functions:

1. Enable users to create secure accounts and log daily symptoms including menstrual cycle information, physical symptoms (acne, hirsutism), mood changes, weight fluctuations, dietary habits, physical activity, and medication compliance.

2. Store all user data locally using SQLite database, ensuring complete user control over personal health information and addressing privacy concerns associated with cloud storage.

3. Generate interactive visualizations using Python libraries (Matplotlib) and JavaScript charting libraries to display symptom trends over time, identify correlations between different variables, and reveal patterns that may not be apparent in raw data.

4. Provide customizable tracking parameters allowing users to add or modify tracked symptoms based on their individual needs and priorities.

5. Enable users to generate comprehensive reports summarizing symptom data over specified time periods, which can be exported in formats suitable for sharing with healthcare providers.

8. Provide an intuitive, user-friendly interface accessible through web browsers on various devices including desktop computers, tablets, and smartphones.

3.5 Proposed System Architecture

The PCOS symptom tracking system follows a three-tier architecture consisting of the presentation layer, application logic layer, and data storage layer. This modular architecture ensures separation of concerns, facilitating maintainability, scalability, and future enhancements.

Presentation Layer (User Interface): The presentation layer, developed with HTML5, CSS3, and JavaScript, offers a responsive web interface for user interaction. It features login pages, dashboards, data entry forms, visualization panels, and report generation tools.

Application Logic Layer (Backend Processing): The application logic layer, built with Python, manages data processing, analysis, and core business operations. It handles authentication, input validation, statistical analysis, visualization, report generation, and data export (PDF, CSV).

Data Storage Layer (Database): The system uses SQLite as a lightweight, serverless database for local data storage, managing user accounts, symptom logs, menstrual cycle data, weight, and mood records.

3.6 System Design

The process of defining the architecture, interfaces, and data for a system that complies with particular requirements is known as system design. It is a multidisciplinary field that entails trade-off analysis, weighing competing requirements, and making choices regarding design decisions that will have an impact on the entire system. A methodical approach to developing and engineering systems is necessary for system design.

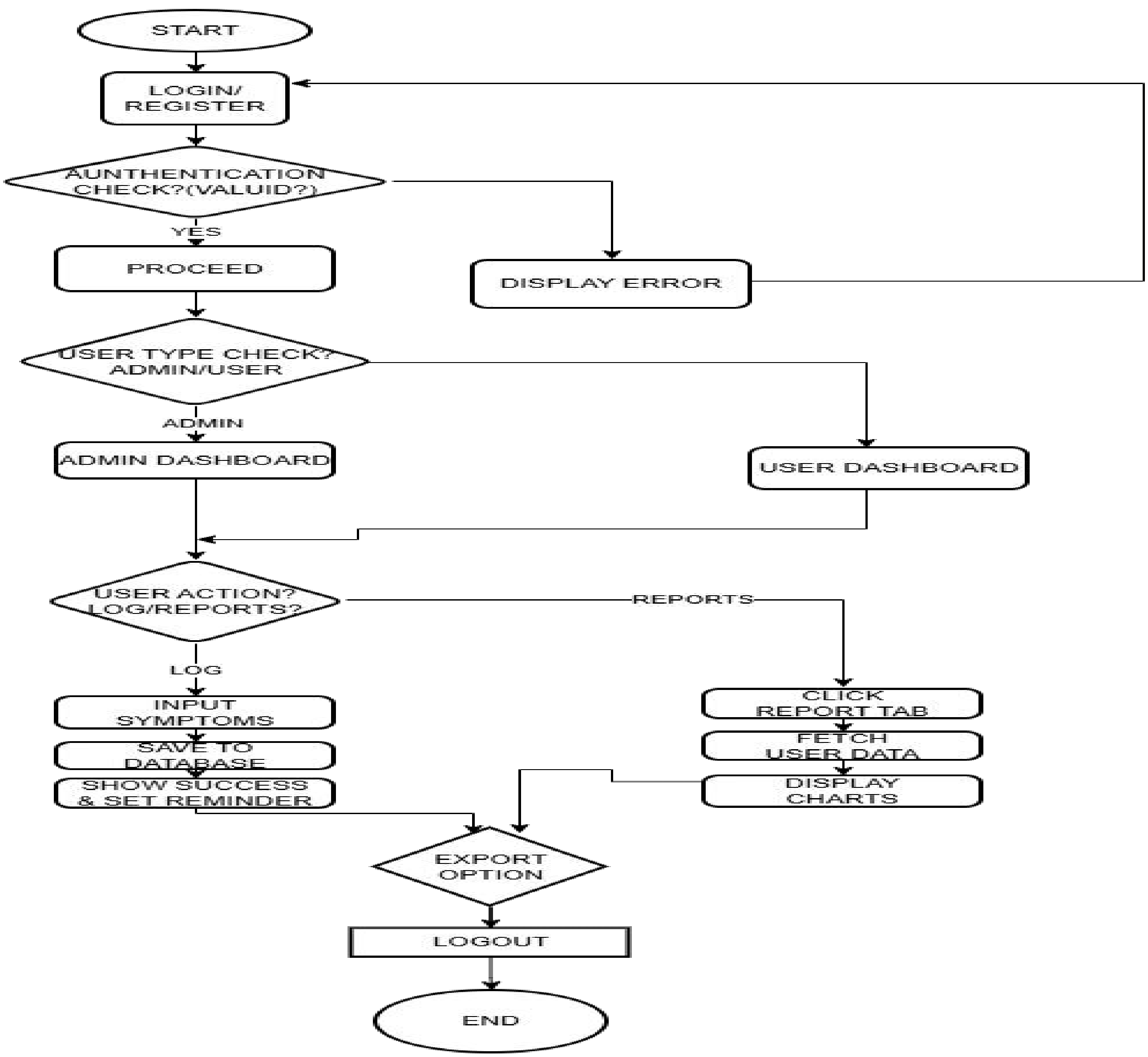

Figure 3.1: Flowchart of the Proposed System

3.7 System Design Tools

The creation of precise plans, diagrams, and specifications for software systems is aided by the use of system design tools. System architectures, components, interactions, and other design elements can be represented visually with the help of these tools. Common types of system design tools include:

1. **Unified Modelling Language (UML**): UML is a standardized language that is used to specify how software systems should be visualized, built, and documented. It is an essential tool for creating object-oriented software and the software development process, and it is independent of any specific programming language or method.

2. **Data Flow Diagram (DFD):** A data flow diagram is a tool used to explain the movement of data through a system and the operations or processing that it carries out. It displays how data enters the system from external sources, how it travels from one process to another, and how it is logically stored.

3. **Entity-Relationship Diagram (ER Diagram)**: This displays the entities and connections that the data describes. It can refer to a group of people, places, things, occasions, or even concepts. These relationships include one-to-one, one-to-many, many-to-many, and many-to-one relationships.

4. **System Flowchart**: The system flow chart is a diagrammatic representation that shows how a system operates; this is easier to understand than reading a long text. One of the main tools a system analyst uses to display a high-level picture of the operations in an entire system is a system flow chart.

3.8 System Design Tool: UML

Unified Modelling Language, also known as UML, is a standardized modelling language that consists of a collection of integrated diagrams. It was created to assist system and software developers in defining, visualizing, building, and documenting the artefacts of software systems as well as business modelling and other non-software systems. Creating object-oriented software and the software development process both heavily rely on the UML. The UML primarily employs graphical notations to convey software project design.

3.9 UML – Use Case Diagram

A use case diagram is a dynamic or behaviour diagram in UML that models the functionality of a system using actors and use cases. The actors are people or entities operating under defined roles within the system. The use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web application.

OBJECT	SYMBOL	DESCRIPTION
Actors	Stick figure	These are the people who interact with the system. An actor is a person, a group, or an external system that interacts with the application or system. In the PCOS tracking system, the primary actor is the woman with PCOS who uses the system to track her symptoms
Use case	Oval/Ellipse	A use case is a collection of

		tasks, services, and operations that the system must carry out. It aids in capturing the system's features and how users interact with it without delving into the system's implementation's internals. Examples include "Log Symptoms," "View Visualizations," and "Generate Report."
System	Rectangle	The term "system" refers to the complete software application in question. It depicts the PCOS symptom tracking and visualization system being developed.
Association	Solid line	Connects an actor to a use case, indicating the actor participates in that use case
Include Relationship.	Dashed arrow labeled «include»	Indicates that one use case always includes the behavior of another use case
Extend Relationship.	Dashed arrow labeled «extend»	Indicates that one use case may optionally extend another use case under certain conditions

Table 3.1 Use Case Notations and Descriptions

Key Use Cases for PCOS Symptom Tracking System:

1. User Registration and Login

   - Actor: User (Woman with PCOS)

   - Description: User creates an account and securely logs into the system

2. Log Daily Symptoms

   - Actor: User

   - Description: User enters daily symptom data including menstrual information, physical symptoms, mood, weight, diet, and activities

   - Includes: Validate Data Input

3. View Symptom History

   - Actor: User

   - Description: User views previously logged symptom data in chronological order

   - Includes: Retrieve Data from Database

4. Generate Visualizations

   - Actor: User

   - Description: User creates interactive charts and graphs showing symptom trends and patterns

   - Includes: Analyze Data, Create Charts

5. Generate Reports

   - Actor: User

- Description: User creates comprehensive reports summarizing symptom data over specified periods

- Includes: Export to PDF/CSV

3.10 UML – State Machine Diagram

A state machine is any device that stores the status of an object at a given time and can change status or cause other actions based on the input it receives. State diagrams mainly depict states and transitions. States are represented with rectangles with rounded corners that are labelled with the name of the state. It illustrates the states and transitions of an object or system.

OBJECTS	SYMBOLS	DESCRIPTION
Initial State	Filled circle	Marks the starting point when an object is created or a process begins
Final State.	Circle with filled inner circleIndicates	Indicates the endpoint where an object is destroyed or a process completes
State	Rounded Rectangle	Indicates the current nature of an object
Transition Arrow	Arroww	An arrow running from one state to another that indicates a changing state
Decision point	Diamond	Indicates a dynamic condition with branched potential results
Fork/join	Thick horizontal bar	Fork splits one transition into

		multiple concurrent paths; join synchronizes multiple paths

Table 3.2: State Machine Diagram Notations and Descriptions

Key States in PCOS Symptom Tracking System:

User Session State Machine:

The user session progresses through distinct states from initial access to logout:

Initial State → Not Authenticated

Not Authenticated → [Login Request] → Authenticating

Authenticating → [Valid Credentials] → Active Session

Authenticating → [Invalid Credentials] → Authentication Failed → Not Authenticated

Active Session → [User Activity] → Active Session (remains active during use)

Active Session → [Inactivity Timeout] → Session Expired → Not Authenticated

Active Session → [Logout Request] → Logging Out → Final State

Symptom Record State Machine:

Individual symptom records transition through states during creation and modification:

Initial State → New Record

New Record → [User Input] → Being Edited

Being Edited → [Validation] → Decision Point

Decision Point → [Data Valid] → Pending Save

Decision Point → [Data Invalid] → Validation Error → Being Edited

Pending Save → [Save Request] → Saving to Database

Saving to Database → [Save Successful] → Saved

Saving to Database → [Save Failed] → Save Error → Pending Save

Saved → [Edit Request] → Being Edited

Saved → [Delete Request] → Deleted → Final State

Visualization Generation State Machine:

Visualizations progress through processing stages from user request to display:

Initial State → Visualization Requested

Visualization Requested → [Fetch Data] → Retrieving Data

Retrieving Data → [Analysis] → Analyzing Data

Analyzing Data → [Sufficient Data] → Generating Chart

Analyzing Data → [Insufficient Data] → Error: Insufficient Data → Final State

Generating Chart → [Render] → Rendering Visualization

Rendering Visualization → [Success] → Visualization Displayed

Rendering Visualization → [Error] → Render Error → Final State

Visualization Displayed → [New Request] → Visualization Requested

3.11 UML – Class Diagram

One of the more popular types in UML is the class diagram, popular among software engineers to document software architecture. Class diagrams are a type of structure diagram because they describe what must be present in the system being modelled.

A Class Diagram provides a visual representation of the system's classes, their attributes, and relationships. In the PCOS symptom tracking system, the following classes can be identified:

OBJECTS	SYMBOLS	DESCRIPTION
Class	Rectangle with three sections	Top section contains class name, middle section lists attributes, bottom section lists methods
Association	A solid connecting two classes	Indicates a relationship
Aggregation	Line with hollow diamond	Indicates "has-a" relationship where one class contains another, but both can exist independently
Composition	Line filled with diamond	Indicates ownership
Dependency	Dashed arrow	Indicates a relationship between two classes where changes in one class might affect the other class
Inheritance	Line with hollow triangle arrow	Points to the parent class

Table 3.3: Class Diagram Notations and Descriptions

Key Classes in PCOS Symptom Tracking System:

1. User Class

- Attributes: userId, email, password, dateCreated, lastLogin

- Methods: login(), logout(), updateProfile(), changePassword()

2. SymptomRecord Class

- Attributes: recordId, userId, date, menstrualCycle, weight, mood, physicalSymptoms, notes

- Methods: createRecord(), updateRecord(), deleteRecord(), validateData()

3. Visualization Class

- Attributes: chartId, userId, chartType, dataRange, parameters

- Methods: generateLineChart(), generateBarChart(), generateHeatmap(), exportChart()

4. Report Class

- Attributes: reportId, userId, startDate, endDate, format

- Methods: generateReport(), exportToPDF(), exportToCSV()

5. Database Class

- Attributes: dbConnection, dbPath

- Methods: connect(), disconnect(), executeQuery(), backup(), restore()

3.12 Database Schema Design

The database schema for the PCOS symptom tracking system is designed to efficiently store and retrieve user data while maintaining data integrity and supporting complex queries for analysis and visualization.

Key Database Tables:

1. Users Table

- user_id (Primary Key)

- email (Unique)

- password_hash

- created_at

- last_login

2. Symptom_Records Table

- record_id (Primary Key)

- user_id (Foreign Key)

- record_date

- weight

- mood_rating

- acne_severity

- hirsutism_level

- created_at

3. Menstrual_Cycles Table

- cycle_id (Primary Key)

- user_id (Foreign Key)

- start_date

- end_date

- flow_intensity

- cycle_length

The database design ensures data normalization to minimize redundancy while maintaining referential integrity through foreign key constraints. Indexes are created on frequently queried fields to optimize performance.

CHAPTER FOUR

SYSTEMS DESIGN AND IMPLEMENTATION

4.1 Software Implementation Tools

The successful development and deployment of Esosa's PCOS Tracker required a careful blend of frontend, backend, database, and visualization tools. Each tool was chosen to achieve reliability, responsiveness, and be user friendly while remaining lightweight enough for deployment.

4.1.1 Frontend Tools

1. **HTML5**: Provided the basic structure of all web pages including the Register, Login, Dashboard, and Reports pages. Semantic tags ensured accessibility and search-engine readability.

2. **CSS**: Handled layout, color, and responsiveness across screen sizes. Media queries were implemented to adapt the interface for mobile devices, tablets, and desktops, ensuring users on iPhones and Android phones experienced consistent visuals.

3. **JavaScript (Vanilla JS):** Allowed the client to interact with the server properly. The scripts handled symptom form validation, requests to the Flask backend, and live updates of charts and summary cards.

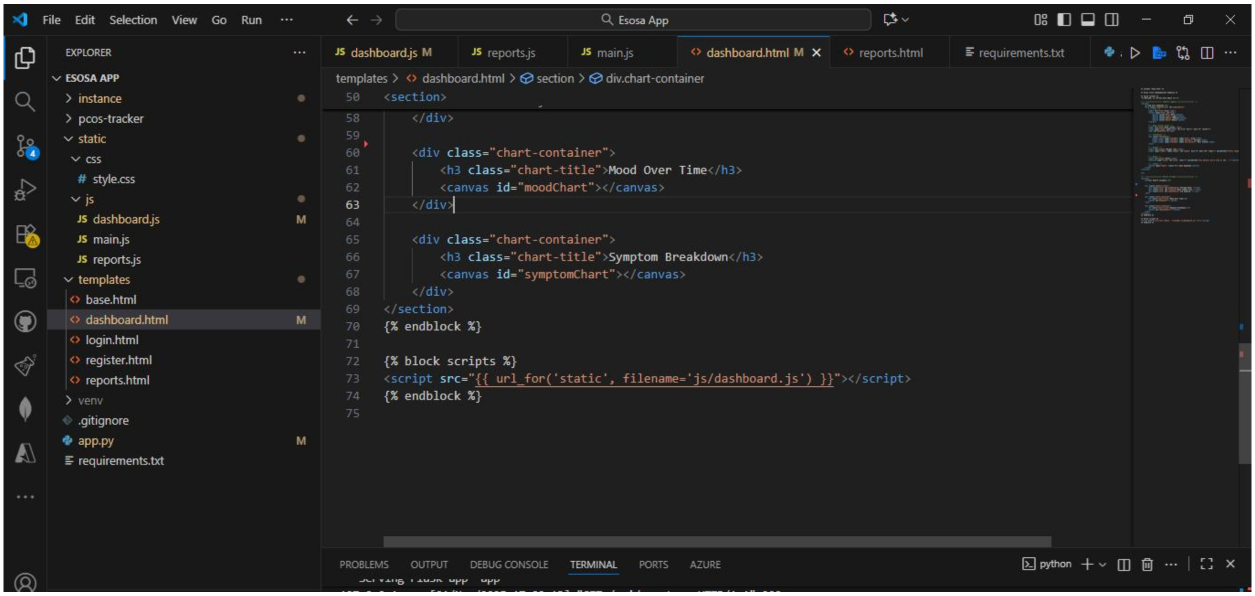

Figure 4.1: Screenshot of Folder Structure Showing HTML Templates and JS Files

4.1.2 Backend Tools

Python: Served as the primary backend programming language. Python's simplicity allowed quick prototyping of data processing and visualization logic.

Flask: A lightweight Python web framework that handled all routing, user authentication, and communication between frontend and backend. Flask's modular design supported RESTful APIs for logging symptoms, retrieving analytics data, and downloading CSV/PDF reports.

Flask-CORS: Enabled secure cross-origin requests during development and testing, allowing the frontend and backend to communicate seamlessly even when hosted separately.

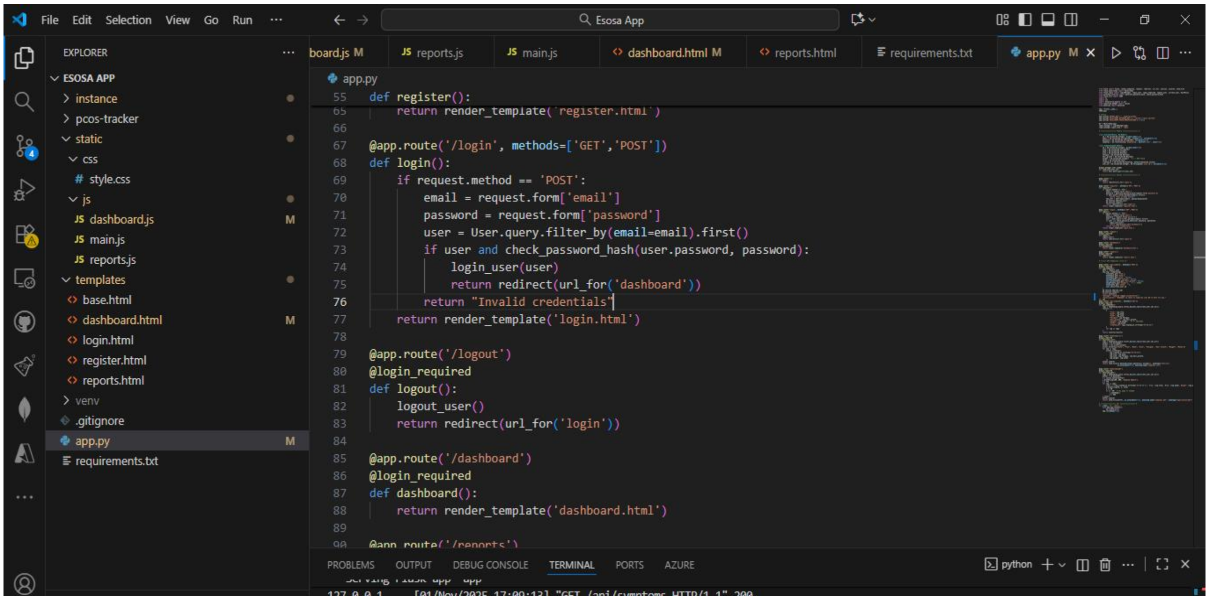

Figure 4.2: Screenshot of Flask App Structure Showing Routes

4.1.3 Database Tools

• **SQLite**: Used as the primary relational database during local development. It stored user accounts, daily symptom logs, and weight entries in a lightweight `.sqlite3` file within the project's instance directory.

• **PostgreSQL (Production):** Selected as the production database for deployment on Render. It offered better scalability, persistent storage, and compatibility with SQLAlchemy ORM models defined in Flask.

4.1.4 Visualization and Reporting Tools

• **Chart.js**: Provided responsive, client-side visualizations for daily and monthly insights. It was used to render mood-flow correlation charts, symptom frequency pies, and line charts for weight trends.

• **Matplotlib**: Used server-side to generate static visual summaries for PDF downloads, ensuring reports could be shared or printed.

• **ReportLab**: Generated downloadable PDF reports that summarize each user's logged symptoms and analytics over time.

4.2 User Documentation

4.2.1 System Testing

System testing evaluated the complete PCOS Tracker application to ensure every module interacted correctly and met functional requirements. The following features were tested:

1. Successful user registration and authentication using email and password.
2. Proper dashboard display after login, showing summary cards and symptom form.
3. Accurate logging of flow, mood, symptoms, notes, and weight.
4. Correct storage and retrieval of data from SQLite/PostgreSQL databases.
5. Real-time visualization updates after each form submission.
6. Correct rendering of charts (bar, pie, and line) on both desktop and mobile.

7. Successful generation and download of CSV and PDF reports.

4.2.2 Integration Testing

Integration testing verified the flow between frontend, backend, and database:

1. User logs symptoms from the dashboard.
2. Flask API stores data in the database.
3. Reports page fetches and visualizes aggregated data using Chart.js.
4. Download buttons export summaries via Flask’s file-handling functions.

All integrations were confirmed to execute smoothly under normal network conditions.

4.2.3 System Testing

System-wide testing confirmed end-to-end usability:

- Forms displayed correct feedback upon submission.
- Charts resized correctly across browsers.
- PDF downloads retained chart clarity and included all recorded metrics.
- No broken routes or missing assets during deployment.

4.2.4 User Testing

Ten female students interacted with the prototype over one week. They were asked to:

• Register and log in daily.
• Submit flow, mood, and symptom data.
• Check their dashboard and download a weekly report.

Feedback was majorly positive, highlighting ease of use, attractive layout, and informative charts. Users suggested adding health tips in future updates.

4.3 Screenshots of the Running System

4.3.1 Register Page

A screenshot displays the registration form allowing users to sign up with an email and password.

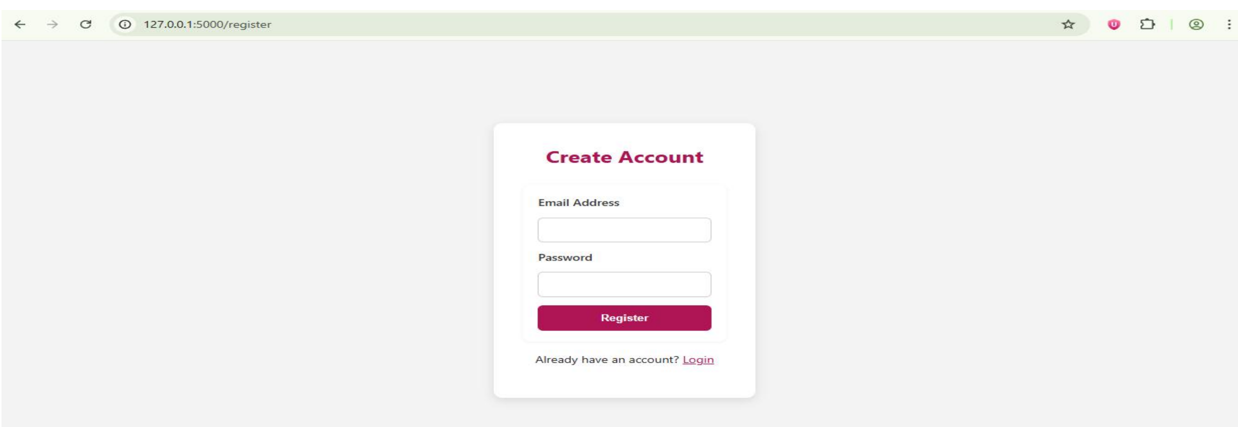

Figure 4.3: Screenshot of Register Page

4.3.2 Login Page

A screenshot shows the login form that redirects users to their dashboard upon successful validation.

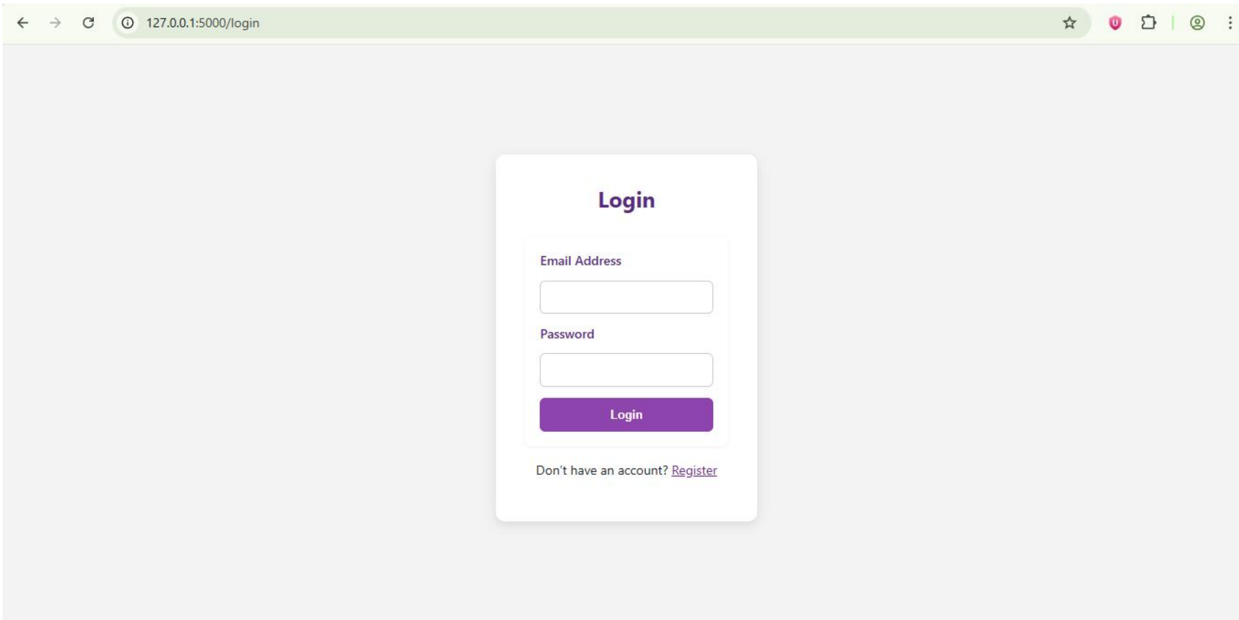

Figure 4.4: Screenshot of Login Page

4.3.3 Dashboard Page

A screenshot shows the main dashboard with symptom logging fields like Flow, Mood, Weight, Notes, as well as notifications.

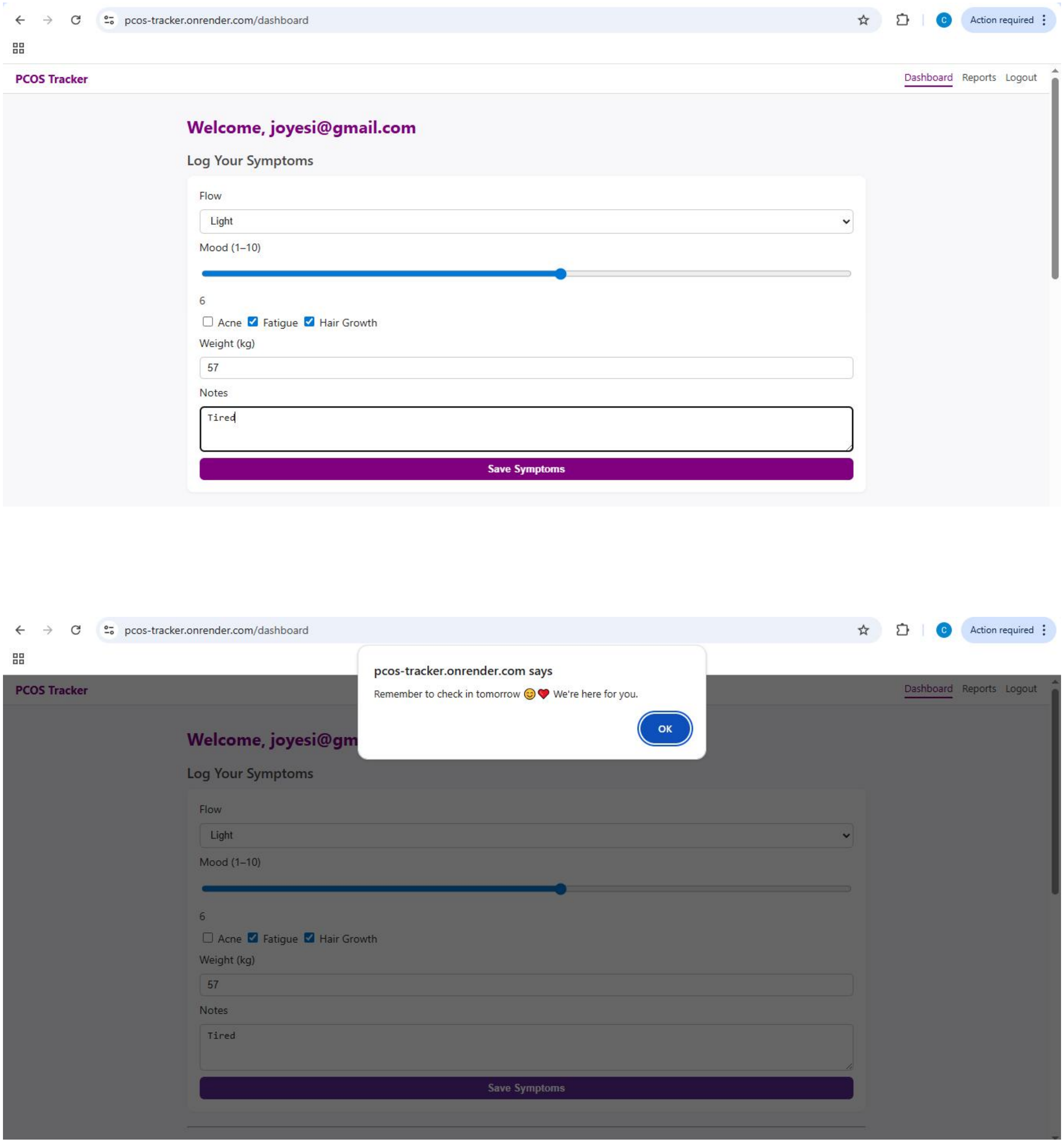

Figure 4.5a: Dashboard Interface on Desktop

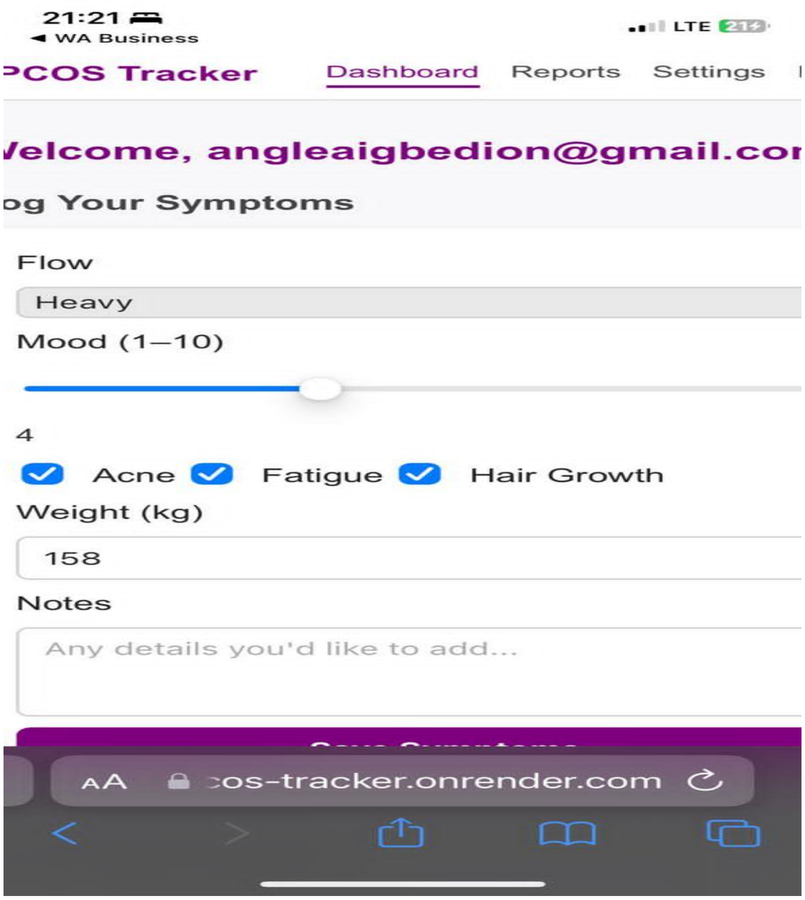

Figure 4.5b: Dashboard Interface on Mobile

4.3.4 Reports Page

Screenshots display interactive charts to show monthly mood and flow trends, symptom frequency breakdown, lifestyle correlations, and weight trend visualization.

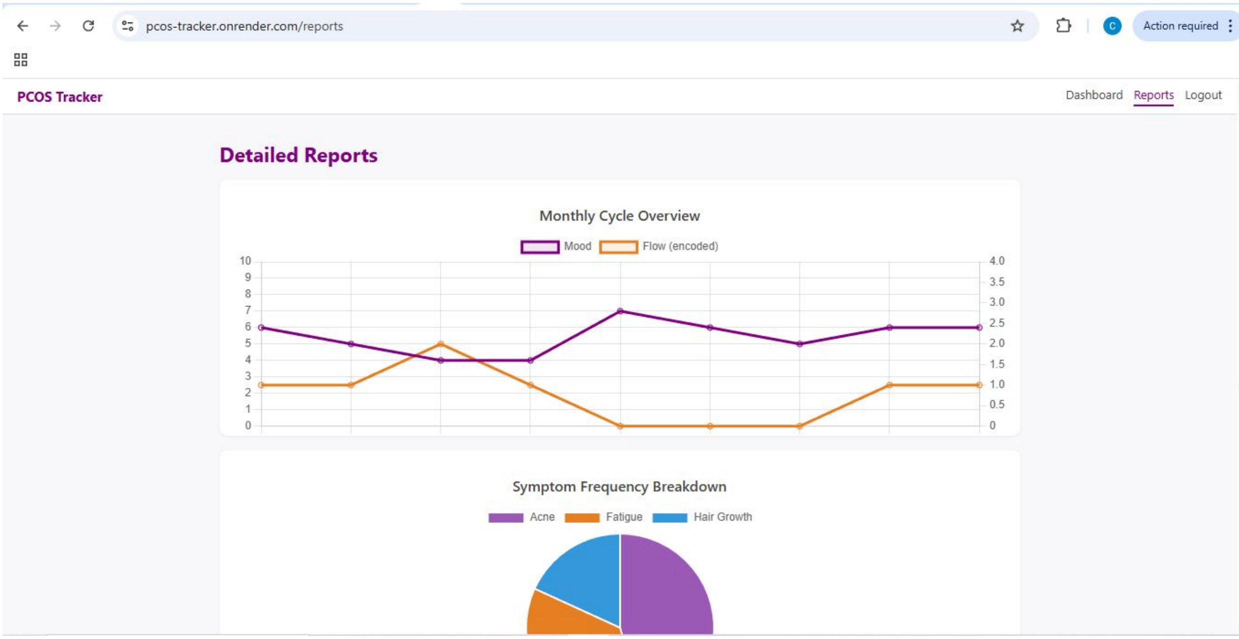

Figure 4.6a: Monthly Cycle Overview Chart

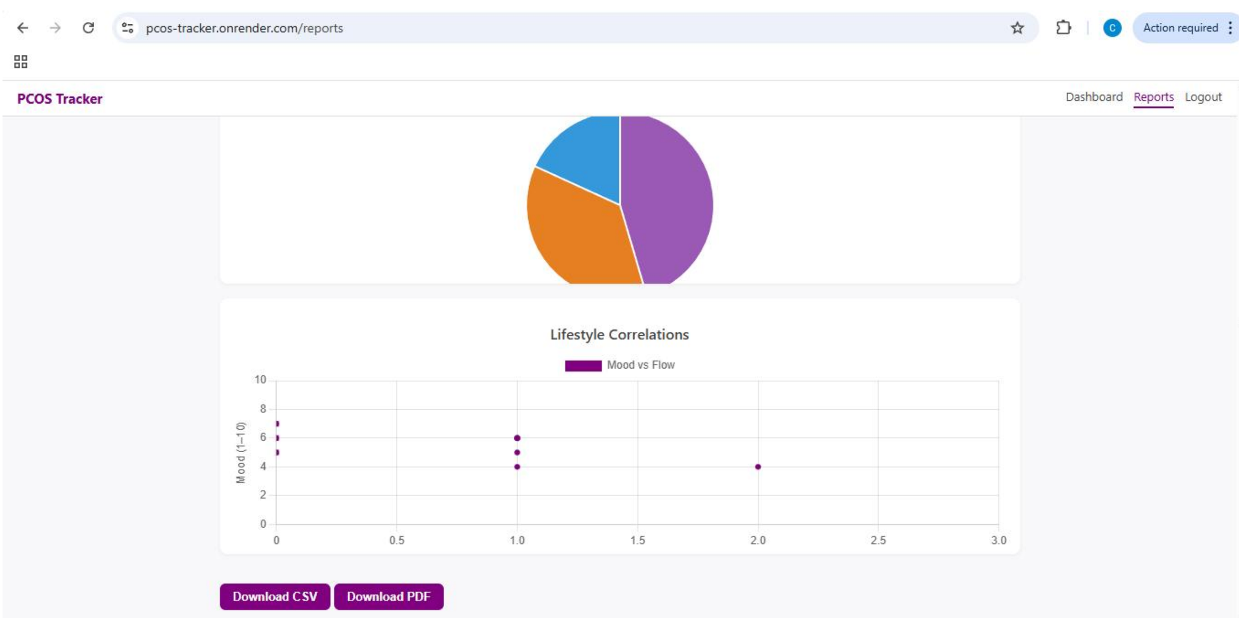

Figure 4.6b: Symptom Frequency Pie Chart and Weight Line Chart with the "Download" Buttons.

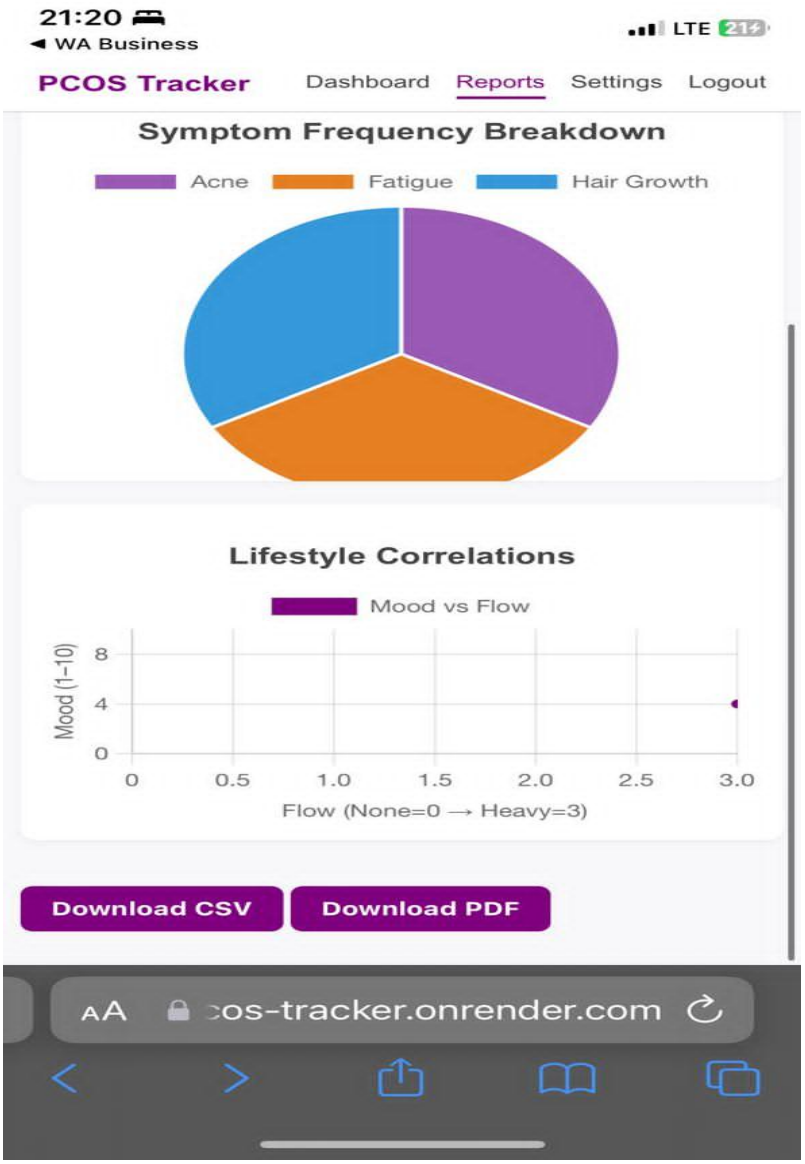

Figure 4.6c: Reports Page on Mobile.

4.3.5 Download Features

Screenshots illustrate working "Download as PDF" and "Download as CSV" buttons.

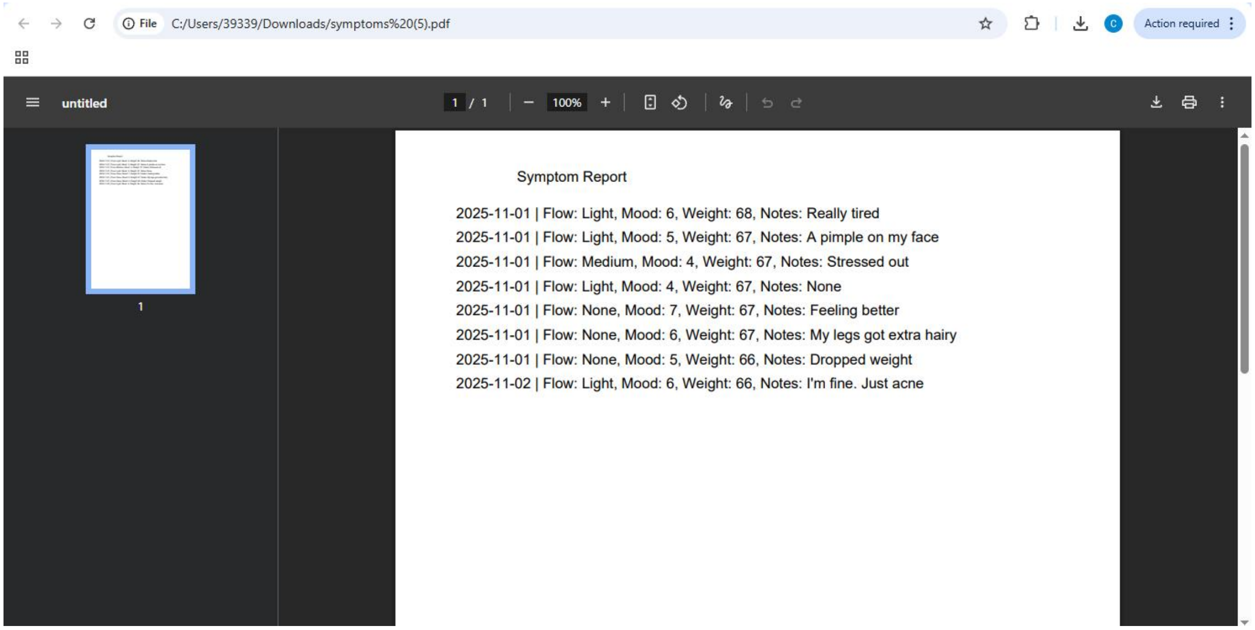

Figure 4.7a: Screenshot of a PDF Report

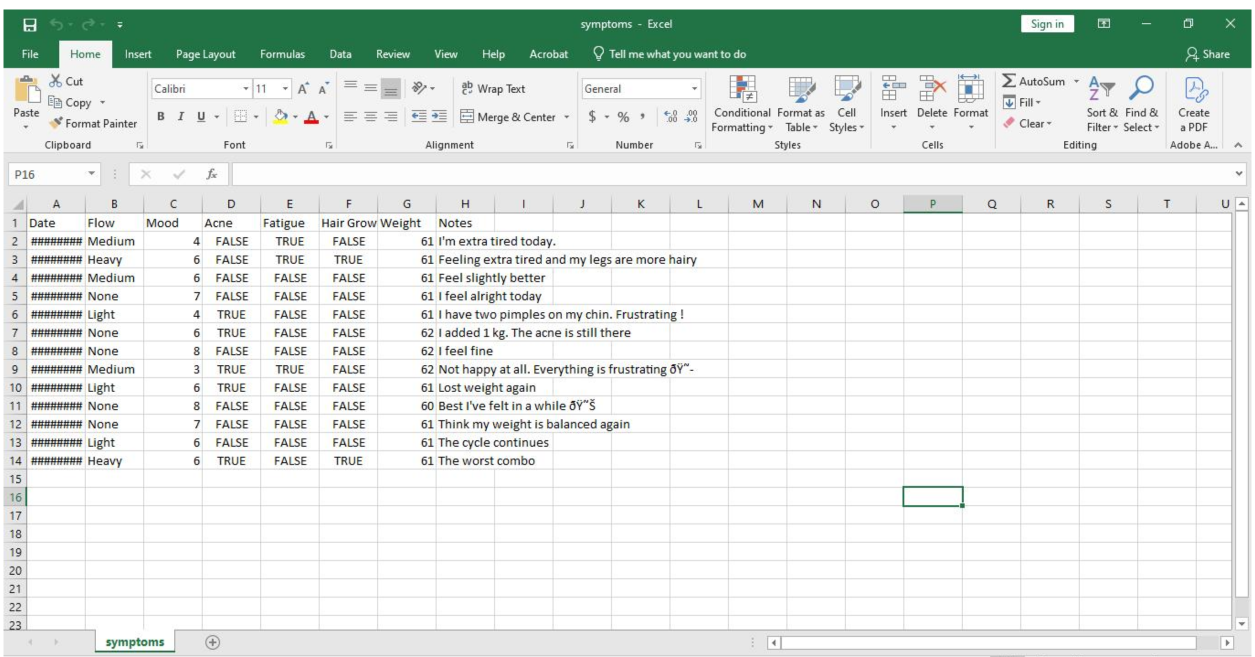

Figure 4.7b: Screenshot of a CSV Report

4.4 System Usability Evaluation

System usability was assessed via a structured questionnaire using Google Forms shared to the test participants. Responses were collected using a 5-point Likert scale ranging from "Strongly Disagree" to "Strongly Agree." The scale includes options:

1. Strongly Disagree.
2. Disagree.
3. Neutral.
4. Agree.
5. Strongly Agree.

4.4.1 Evaluation Metrics

1. **Easy Usage**: How intuitive users found the interface.
2. **Responsiveness**: How quickly charts and pages loaded after data submission.
3. **Data Accuracy**: Whether logged symptoms and weight values were correctly displayed in the reports.
4. **Visual Appeal**: Aesthetic perception of colors, spacing, and layout.
5. **Functionality**: Ability to use the system without lags or breaks.

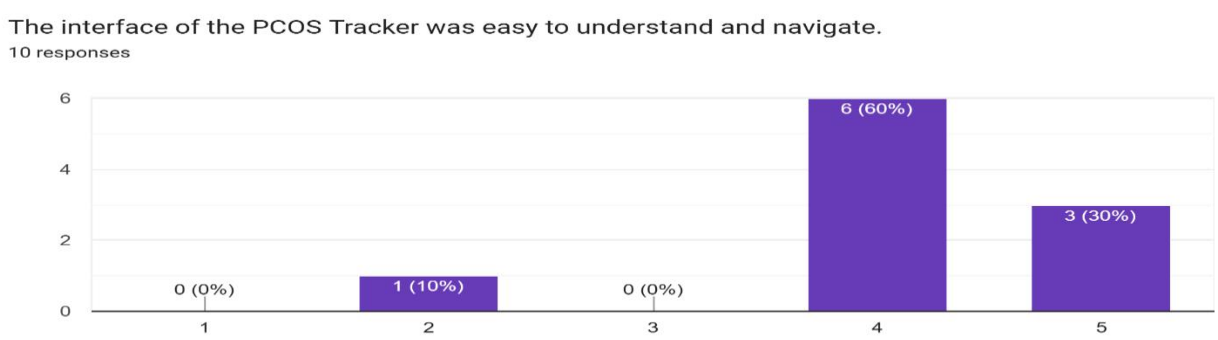

Figure 4.8a: Survey Results Representing Ease of Use

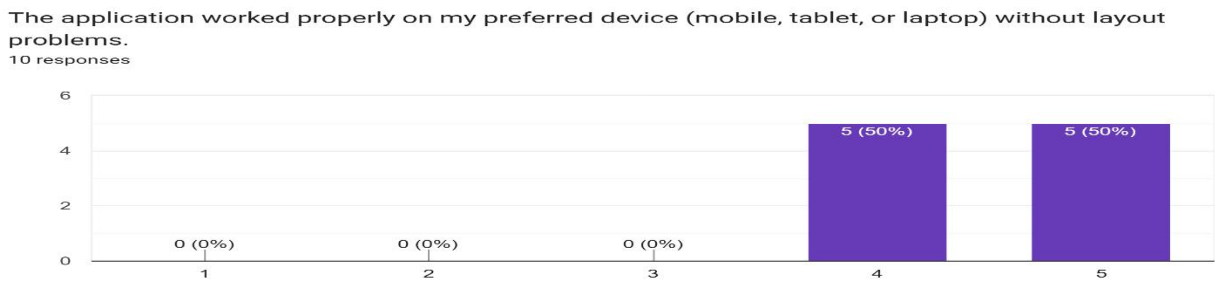

Figure 4.8b: Survey Results Representing Responsiveness

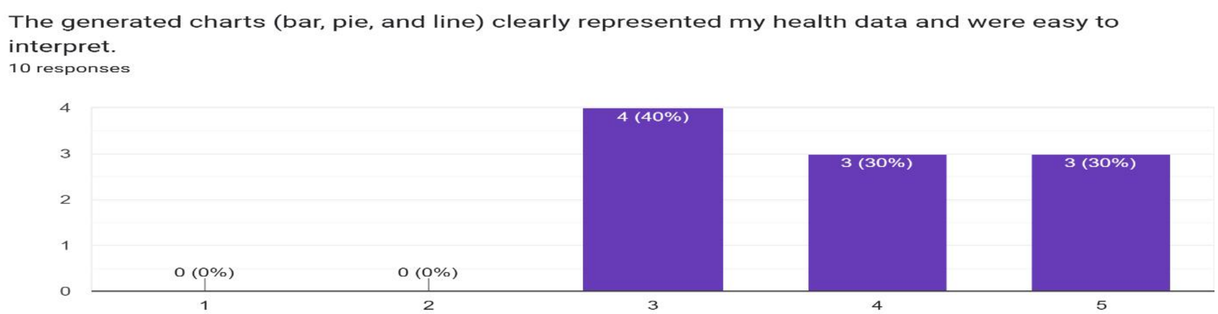

Figure 4.8c: Survey Results Representing Data Interpretation and Accuracy

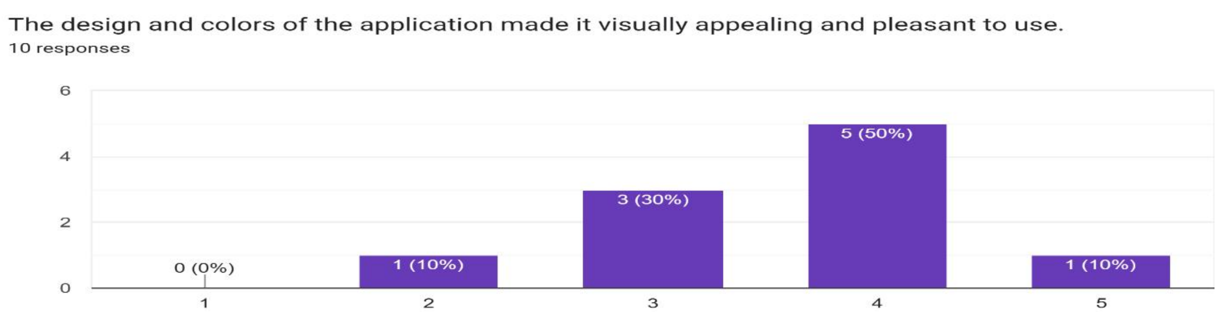

Figure 4.8d: Survey Results Representing Visual Appeal

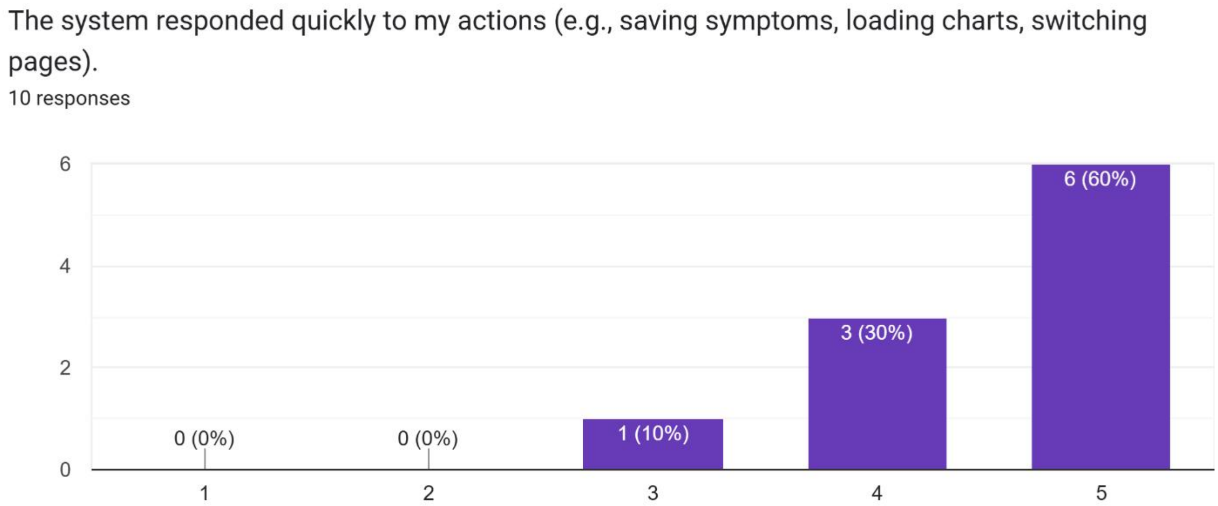

Figure 4.8e: Survey Results Representing System Functionality.

4.4.2 Results

Survey highlights:

• 90% of users agreed the application was easy to use.

• 60% confirmed that visual reports were pleasing to them.

• 100% were satisfied with chart responsiveness on mobile.

• 60% of users reported the data they input was displayed accurately.

• 90% of users agreed the system was fast and functional.

These findings confirm that Esosa's PCOS Tracker achieved its intended goal of providing an accessible, data-driven health-monitoring platform for women.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

At the moment, PCOS burdens up to one in five reproductive-aged women worldwide with considerable challenges in its diagnosis and long-term management. The chronic nature of PCOS, combined with diverse symptoms from menstrual irregularities and infertility to metabolic complications and psychological distress and proactive self-management. Effective management requires consistent monitoring, but traditional paper-based tracking is inadequate, as it is prone to being lost, difficult to analyze, and incapable of revealing symptom patterns. This project directly addressed these gaps by developing an integrated, Python-based symptom tracking and visualization system. Leveraging modern web technologies and Python's data analysis capabilities, the project aimed to create an accessible, user-friendly, and privacy-focused solution for women with PCOS.

The development was guided by the Object-Oriented Analysis and Design (OOAD) methodology, progressing through systematic analysis, design, implementation, and evaluation. An extensive literature review first identified critical shortcomings in existing applications, including limited symptom tracking, poor data visualization, and privacy concerns with cloud storage. The resulting system was designed to overcome these flaws by enabling comprehensive logging of a wide array of PCOS-related symptoms, such as menstrual patterns, weight, mood, and medication compliance. For the backend data processing and analysis, Python was used, while the frontend interface was built with HTML, CSS, and JavaScript to ensure accessibility for users with varying technical skills.

Key technical achievements include the integration of interactive data visualizations, created with JavaScript and Python's Matplotlib, which provide users with line graphs for trend analysis, bar charts for symptom frequency, and dashboard overviews. A paramount feature is the emphasis on privacy and security, achieved by using a local SQLite database that stores all sensitive health data solely on the user's device. Furthermore, the system includes a reporting feature that allows users to generate and export summarized reports for sharing with healthcare providers, thereby facilitating more informed consultations and treatment decisions.

Deployed as a web application on Render, the system is accessible across all devices via a browser, eliminating the need for native mobile apps and ensuring wide accessibility while simplifying maintenance. Usability testing yielded highly positive feedback, confirming that the system is easy to use, provides comprehensive tracking, and features clear visualizations. Test participants particularly valued the privacy assurance of local data storage. The final system successfully meets its objectives, offering a practical tool that empowers women in the proactive self-management of PCOS.

5.2 Conclusion

This research work successfully illustrated the use of Python-based technologies in the development of effective digital health solutions for the management of chronic diseases, such as the unique needs presented by women with PCOS. The developed symptom tracking and visualization system represents an important contribution to health informatics; it offers a practical, accessible, and privacy-oriented tool that empowers women toward active engagement in the management of their condition.

This study confirms that the OOAD methodology works in the development of user-centered healthcare applications by showing that systematic analysis and design processes lead to solutions that meet the users' needs. Further positive usability test results confirm that the balance between functionality and usability has been reached in the system, offering advanced analytical capability through an interface that is accessible for end-users of various technical backgrounds.

The project also illustrates the broader potential of digital health for addressing healthcare challenges, especially in contexts where access to specialized care is limited. The web application-based system, therefore, accessible on all devices without any need for native mobile apps, exemplifies how technology can democratize access to healthcare tools while upholding the privacy and personal autonomy of users.

5.3 Recommendations

Based on the findings and experiences gained in this research project, the following recommendations are proposed for the future work:

1. Health facilities should adopt symptom tracking digital systems by incorporating them into the care of patients and training patients and healthcare providers in their effective use.

2. Future versions of the system should, if possible, integrate machine learning algorithms to provide predictive analytics and personalized treatment recommendations based on individual symptom patterns.

3. The system should be further enhanced by the development of native mobile applications for iOS and Android platforms to increase accessibility, as well as device-specific features like push notifications and offline functionality.

4. Integrations with wearable devices and health tracking platforms should be implemented to enable automated collection of data, reducing manual entry burdens.

5. Clinical validation studies involving larger, diverse populations should be conducted to rigorously evaluate the system's impact on health outcomes and treatment adherence.

6. Developing the system in multiple languages will significantly expand the system's reach to a wide range of global populations, removing language barriers to accessing effective management tools for PCOS.

7. Releasing the system as an open-source software would provide the global developer community with the ability to contribute improvements and adapt the solution for various contexts, along with other chronic conditions.

REFERENCES

Alessa, T., Abdi, S., Hawley, M. S., & de Witte, L. (2018). Mobile apps for health: A critical review of the major design and conceptual issues. *JMIR mHealth and uHealth*, *6*(12), e11763.

Arabkermani, R., Johnson, H., & Lee, S. (2025). Evaluating mobile health applications for polycystic ovary syndrome: Gaps and opportunities. *Journal of Women's Digital Health*, *14*(2), 101–115.

Azziz, R., Carmina, E., Chen, Z., Dunaif, A., Laven, J. S., Legro, R. S., … Yildiz, B. O. (2016). Polycystic ovary syndrome. *Nature Reviews Disease Primers*, *2*(1).

Bazarganipour, F., Ziaei, S., Montazeri, A., Foroozanfard, F., Kazemnejad, A., & Faghihzadeh, S. (2018). Psychological investigation in patients with polycystic ovary syndrome. *Health and Quality of Life Outcomes*, *16*(1), 1–11.

Bozdag, G., Mumusoglu, S., Zengin, D., Karabulut, E., & Yildiz, B. O. (2016). The prevalence and phenotypic features of polycystic ovary syndrome: A systematic review and meta-analysis. *Human Reproduction*, *31*(12), 2841–2855.

Clue. (2025). *Clue period & cycle tracker: Menstrual health app*. Clue.

Cooney, L. G., Lee, I., Sammel, M. D., & Dokras, A. (2017). High prevalence of moderate and severe depressive and anxiety symptoms in polycystic ovary syndrome: A systematic review and meta-analysis. *Human Reproduction*, *3*(5), 1075–1091.

Copp, T., Jansen, J., Doust, J., Mol, B. W., Dokras, A., McCaffery, K., & Cvejic, E. (2021). Supporting shared decision-making for diagnostic testing in primary care for polycystic ovary syndrome: Development of a patient decision aid. *BMC Medical Informatics and Decision Making*, *21*(1), 1–12.

Deswal, R., Narwal, V., Dang, A., & Pundir, C. S. (2020). The prevalence of polycystic ovary syndrome: A brief systematic review. *Journal of Human Reproductive Sciences*, *13*(4), 261–271.

Dokras, A., Clifton, S., Futterweit, W., & Wild, R. (2011). Increased risk for abnormal depression scores in women with polycystic ovary syndrome: A systematic review and meta-analysis. *Obstetrics & Gynecology*, *117*(1), 145–152.

Doraiswamy, S., Abraham, A., Mamtani, R., & Cheema, S. (2020). Use of telehealth during the COVID-19 pandemic: Scoping review. *Journal of Medical Internet Research*, *22*(12), e24087.

Goodarzi, M. O., Dumesic, D. A., Chazenbalk, G., & Azziz, R. (2011). Polycystic ovary syndrome: Etiology, pathogenesis and diagnosis. *Nature Reviews Endocrinology*, *7*(4), 219–231.

Harlow, S. D., & Ephross, S. A. (1995). Epidemiology of menstruation and its relevance to women's health. *Epidemiologic Reviews*, *17*(2), 265–286.

Jain, T., Negris, O., Brown, D., Galic, I., Salimgaraev, R., & Zhaunova, L. (2020). *Characterization of polycystic ovary syndrome among Flo app users around the world*. National Institute of Medicine.

Legro, R. S., Arslanian, S. A., Ehrmann, D. A., Hoeger, K. M., Murad, M. H., Pasquali, R., & Welt, C. K. (2013). Diagnosis and treatment of polycystic ovary syndrome: An Endocrine Society clinical practice guideline. *The Journal of Clinical Endocrinology & Metabolism*, *98*(12), 4565–4592.

Medling, A. (2018). *Healing PCOS: A 21-day plan for reclaiming your health and life with polycystic ovary syndrome*. HarperOne.

Rodriguez, A., Patel, V., & Singh, R. (2020). Mobile health applications for early detection of polycystic ovary syndrome. *Journal of Women's Health Technology*, *14*(2), 5.

Solat, S., Khan, A., & Yusuf, R. (2025). Evaluating mobile health interventions for polycystic ovary syndrome: Opportunities and gaps. *Journal of Digital Health Research*, *12*(1), 45–59.

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

Stone, A. A., Shiffman, S., Schwartz, J. E., Broderick, J. E., & Hufford, M. R. (2000). Patient compliance with paper and electronic diaries. *Controlled Clinical Trials*, *21*(5 Suppl), 200S–209S.

Teede, H. J., Misso, M. L., Costello, M. F., Dokras, A., Laven, J., Moran, L., … International PCOS Network. (2018). Recommendations from the international evidence-based guideline for the assessment and management of polycystic ovary syndrome. *Human Reproduction*, *33*(9), 1602–1618.

Wagner, E. H., Austin, B. T., Davis, C., Hindmarsh, M., Schaefer, J., & Bonomi, A. (2001). Improving chronic illness care: Translating evidence into action. *Health Affairs*, *20*(6), 64–78.

World Health Organization (WHO). (2023). *Noncommunicable diseases: Risk factors*.

Zhou, Y., Zhang, Y., Wang, L., Li, H., & Chen, X. (2024). Digital health platform for chronic disease management: A systematic review and meta-analysis. *Journal of Medical Internet Research*, *26*.

APPENDIX

BASE PAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PCOS Tracker - {% block title %}{% endblock %}</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>


    {% block styles %}{% endblock %}
</head>
<body>
    <header class="navbar">
        <div class="logo">
            <span>PCOS Tracker</span>
        </div>
        <nav>
            <a href="{{ url_for('dashboard') }}" class="{% if request.endpoint == 'dashboard' %}active{% endif %}">Dashboard</a>
```

```
            <a href="{{ url_for('reports') }}" class="{% if request.endpoint == 'reports' %}active{% endif %}">Reports</a>

            <a href="#">Settings</a>

            <a href="{{ url_for('logout') }}">Logout</a>

        </nav>

    </header>



    <main class="container">

        {% block content %}{% endblock %}

    </main>



    <script src="{{ url_for('static', filename='js/main.js') }}"></script>

    {% block scripts %}{% endblock %}

</body>

</html>
```

REGISTER PAGE

```
{% extends "base.html" %}



{% block title %}Register{% endblock %}



{% block content %}
```

```
<body class="auth-page register-page">
    <div class="auth-container">
        <h1>Create Account</h1>
        <form method="POST" action="{{ url_for('register') }}" class="auth-form">
            <label for="email">Email Address</label>
            <input type="email" name="email" required>

            <label for="password">Password</label>
            <input type="password" name="password" required>

            <button type="submit" class="btn">Register</button>
        </form>
        <p>Already have an account? <a href="{{ url_for('login') }}">Login</a></p>
    </div>
</body>
{% endblock %}
```

LOGIN PAGE

```
{% extends "base.html" %}

{% block title %}Login{% endblock %}
```

```
{% block content %}
<body class="auth-page login-page">
  <div class="auth-container">
    <h1>Login</h1>
    <form method="POST" action="{{ url_for('login') }}" class="auth-form">
      <label for="email">Email Address</label>
      <input type="email" name="email" required>

      <label for="password">Password</label>
      <input type="password" name="password" required>

      <button type="submit" class="btn">Login</button>
    </form>
    <p>Don’t have an account? <a href="{{ url_for('register') }}">Register</a></p>
  </div>
</body>
{% endblock %}
```

DASHBOARD PAGE

```
{% extends "base.html" %}

{% block title %}Dashboard{% endblock %}
```

```
{% block content %}
<h1>Welcome, {{ current_user.email }}</h1>
<section>
  <h2>Log Your Symptoms</h2>
  <form class="symptom-form" id="symptomForm">

    <label for="flow">Flow</label>
    <select name="flow" id="flow">
      <option value="None">None</option>
      <option value="Light">Light</option>
      <option value="Medium">Medium</option>
      <option value="Heavy">Heavy</option>
    </select>

    <label for="mood">Mood (1–10)</label>
    <input type="range" name="mood" id="mood" min="1" max="10" value="5">
    <span id="moodValue">5</span>

    <div class="checkboxes">
      <label><input type="checkbox" name="acne"> Acne</label>
```

```html
        <label><input type="checkbox" name="fatigue"> Fatigue</label>

        <label><input type="checkbox" name="hair_growth"> Hair Growth</label>

      </div>


      <label for="weight">Weight (kg)</label>

      <input type="number" name="weight" id="weight" min="30" max="200" step="1"
placeholder="Enter weight in kg">





      <label for="notes">Notes</label>

      <textarea name="notes" id="notes" rows="3" placeholder="Any details you'd like to
add..."></textarea>


      <button type="submit" class="btn">Save Symptoms</button>

    </form>

</section>


<hr>



<section>
```

```
    <h2>Your Health Insights</h2>



    <div class="summary-cards">
        <div class="card" id="avgMoodCard">Average Mood: --</div>
        <div class="card" id="flowCard">Most Common Flow: --</div>
        <div class="card" id="symptomCard">Top Symptom: --</div>
    </div>

    <div class="chart-container">
        <h3 class="chart-title">Mood Over Time</h3>
        <canvas id="moodChart"></canvas>
    </div>

    <div class="chart-container">
        <h3 class="chart-title">Symptom Breakdown</h3>
        <canvas id="symptomChart"></canvas>
    </div>
</section>
{% endblock %}
```

```
{% block scripts %}
<script src="{{ url_for('static', filename='js/dashboard.js') }}"></script>
{% endblock %}
```

REPORTS PAGE

```
{% extends "base.html" %}

{% block title %}Reports{% endblock %}

{% block content %}
<h1>Detailed Reports</h1>

<section class="chart-container">
   <h2 class="chart-title">Monthly Cycle Overview</h2>
   <canvas id="cycleChart"></canvas>
</section>
<section class="chart-container">
   <h2 class="chart-title">Symptom Frequency Breakdown</h2>
   <canvas id="symptomPieChart"></canvas>
</section>
<section class="chart-container">
   <h2 class="chart-title">Lifestyle Correlations</h2>
```

```
    <canvas id="lifestyleChart"></canvas>
</section>
<div style="margin-top: 2rem;">
    <button class="btn" id="downloadCsv">Download CSV</button>
    <button class="btn" id="downloadPdf">Download PDF</button>
</div>
{% endblock %}


{% block scripts %}
<script src="{{ url_for('static', filename='js/reports.js') }}"></script>
{% endblock %}
```

PYTHON CODE

```
from flask import Flask, render_template, request, redirect, url_for, session, jsonify, send_file
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager, login_user, login_required, logout_user, current_user, UserMixin
from werkzeug.security import generate_password_hash, check_password_hash
from flask_cors import CORS
```

```python
import io
import os
import csv
import matplotlib.pyplot as plt
from reportlab.pdfgen import canvas
from datetime import datetime

app = Flask(__name__)
CORS(app)

app.config['SECRET_KEY'] = os.getenv("SECRET_KEY", "supersecretkey")
database_url = os.getenv("DATABASE_URL", "sqlite:///app.db")
if database_url.startswith("postgres://"):
    database_url = database_url.replace("postgres://", "postgresql://", 1)

app.config["SQLALCHEMY_DATABASE_URI"] = database_url
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

db = SQLAlchemy(app)
login_manager = LoginManager(app)
login_manager.login_view = 'login'
```

```python
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(150), unique=True, nullable=False)
    password = db.Column(db.String(200), nullable=False)
    symptoms = db.relationship('SymptomLog', backref='user', lazy=True)


class SymptomLog(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    flow = db.Column(db.String(50))
    mood = db.Column(db.Integer)
    acne = db.Column(db.Boolean)
    fatigue = db.Column(db.Boolean)
    hair_growth = db.Column(db.Boolean)
    weight = db.Column(db.Integer)
    notes = db.Column(db.Text)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
```

```
@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))


@app.route('/')
def home():
    return redirect(url_for('login'))


@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        email = request.form['email']
        password = generate_password_hash(request.form['password'])
        if User.query.filter_by(email=email).first():
            return "User already exists!"
        new_user = User(email=email, password=password)
        db.session.add(new_user)
        db.session.commit()
```

```python
        return redirect(url_for('login'))
    return render_template('register.html')


@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        user = User.query.filter_by(email=email).first()
        if user and check_password_hash(user.password, password):
            login_user(user)
            return redirect(url_for('dashboard'))
        return "Invalid credentials"
    return render_template('login.html')


@app.route('/logout')
@login_required
def logout():
    logout_user()
```

```
    return redirect(url_for('login'))


@app.route('/dashboard')
@login_required
def dashboard():
    return render_template('dashboard.html')


@app.route('/reports')
@login_required
def reports():
    return render_template('reports.html')

@app.route('/api/symptom', methods=['POST'])
@login_required
def log_symptom():
    data = request.json
    new_log = SymptomLog(
        flow=data.get('flow'),
        mood=data.get('mood'),
```

```
        acne=data.get('acne', False),
        fatigue=data.get('fatigue', False),
        hair_growth=data.get('hair_growth', False),
        weight=data.get('weight'),
        notes=data.get('notes', ''),
        user_id=current_user.id
    )
    db.session.add(new_log)
    db.session.commit()
    return jsonify({
    "message": "Symptom logged successfully!",
    "notification": "Remember to check in tomorrow 😊♡ We're here for you."
}), 201


@app.route('/api/symptoms', methods=['GET'])
@login_required
def get_symptoms():
    logs = SymptomLog.query.filter_by(user_id=current_user.id).all()
    results = [
        {
```

```
            "flow": log.flow,
            "mood": log.mood,
            "acne": log.acne,
            "fatigue": log.fatigue,
            "hair_growth": log.hair_growth,
            "weight": log.weight,
            "notes": log.notes,
            "created_at": log.created_at.strftime("%Y-%m-%d")
        }
        for log in logs
    ]
    return jsonify(results)


@app.route('/download/csv')
@login_required
def download_csv():
    logs = SymptomLog.query.filter_by(user_id=current_user.id).all()
    output = io.StringIO()
    writer = csv.writer(output)
```

```
    writer.writerow(["Date", "Flow", "Mood", "Acne", "Fatigue", "Hair Growth", "Weight", "Notes"])
    for log in logs:
        writer.writerow([
            log.created_at.strftime("%Y-%m-%d"),
            log.flow, log.mood,
            log.acne, log.fatigue, log.hair_growth,
            log.weight, log.notes
        ])
    output.seek(0)
    return send_file(io.BytesIO(output.getvalue().encode()), mimetype="text/csv",
                     as_attachment=True, download_name="symptoms.csv")


@app.route('/download/pdf')
@login_required
def download_pdf():
    logs = SymptomLog.query.filter_by(user_id=current_user.id).all()
    buffer = io.BytesIO()
    c = canvas.Canvas(buffer)
    c.drawString(100, 800, "Symptom Report")
```

```
    y = 770
    for log in logs:
        line = f"{log.created_at.strftime('%Y-%m-%d')} | Flow: {log.flow}, Mood: {log.mood}, Weight: {log.weight}, Notes: {log.notes}"
        c.drawString(50, y, line)
        y -= 20
        if y < 50:
            c.showPage()
            y = 800
    c.save()
    buffer.seek(0)
    return send_file(buffer, as_attachment=True, download_name="symptoms.pdf", mimetype="application/pdf")


with app.app_context():
    db.create_all()


if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=int(os.getenv("PORT", 5000)))
```