ADROID Conference Series: Engineering and Technology
Volume 2, Issue 3, pp.204-228, 2026

HELIXCARE: A DNA-based Health Prediction and Drug Recommendation System

Cevin R Christo, Divyanshi Kochar, Apurv Bhardwaj
Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, Chennai, India
RA2211042010002@srmist.edu.in, RA2211042010044@srmist.edu.in, RA2211042010050@srmist.edu.in

Abstract. Modern medicine often assumes uniform drug responses, yet genetic variations, especially Single Nucleotide Polymorphisms (SNPs), significantly influence treatment outcomes, leading to adverse reactions and inefficiencies. HelixCare is a web-based precision medicine platform designed to bridge this gap by integrating genomic, clinical and lifestyle data to enable personalised healthcare decisions. The system accepts SNP data (VCF/CSV) along with attributes such as age, BMI, and lifestyle factors, and generates disease risk predictions using an ensemble Random Forest model trained on 106 features. It achieves 87.3% accuracy and an AUC of 0.923, outperforming traditional models including Logistic Regression, SVM, MLP and XGBoost ($p < 0.001$). HelixCare also incorporates a pharmacogenomic recommendation engine that leverages high-evidence drug–gene interactions from PharmGKB, ClinVar and dbSNP to guide safer medication choices, including dose adjustments and drug avoidance. Built on a React-based frontend and a Flask backend, the system ensures efficient, low-latency performance and secure data handling via AES-256 encryption and role-based access control. By enabling accessible, data-driven insights, HelixCare promotes a shift toward proactive, personalised healthcare aligned with global health innovation goals.

Keywords: Precision Medicine, Pharmacogenomics, Single Nucleotide Polymorphism, Random Forest, Ensemble Learning, Drug-Gene Interaction, PharmGKB, Clinical Decision Support, Health Informatics, SDG 3, SDG 9.

1. INTRODUCTION

1.1 Introduction to Project

Healthcare today still treats most patients as if they were average. A doctor writes a prescription, the patient fills it, and everyone hopes the textbook response holds. For a great many patients it does not. People walk out of the same clinic, with the same diagnosis, holding the same pill, and end up with very different outcomes. One person feels better in a week. Another develops a serious side effect. A third sees no improvement at all. The medicine has not changed; the body receiving it has.

These differences are rarely random. They are the visible end of a long chain of biological events that begins with the human genome. Every individual carries small variations in their DNA, called Single Nucleotide Polymorphisms or SNPs, and these tiny variations decide how enzymes break down a drug, how receptors bind to it and how quickly the body removes it from the bloodstream. When a clinician ignores this layer of biology, they are essentially flying blind. The pharmacological knowledge is sound, but it has been stripped of one of its most important inputs.

The cost of ignoring genomics is no longer hypothetical. Adverse drug reactions are now among the leading causes of preventable harm in hospitals worldwide. Studies estimate that between five and seven percent of hospital admissions in developed economies are linked to a drug response that could have been predicted from the patient's genetic profile. The technology to read that profile has, for the first time, become genuinely affordable. A whole-exome sequencing test that cost tens of thousands of dollars a decade ago can now be ordered for the price of a smartphone. The bottleneck has moved from the laboratory to the clinic. The data exists, but very little of it ever reaches the people who need it.

HelixCare is our response to that bottleneck. It is a web-based platform that takes a person's genomic file along with a small set of lifestyle answers and turns them into something a non-specialist can act on. The user uploads either a Variant Call Format (VCF) file or a Comma-Separated Values (CSV) file containing their SNP variants, fills in a short form covering age, body mass index, smoking status, alcohol consumption and physical activity, and within a few seconds receives a personalised dashboard showing their disease risk profile and a list of medications that may need to be adjusted, used cautiously or avoided altogether. Behind that simple interface sits an ensemble Random Forest model trained on a 106-feature vector and a pharmacogenomic engine wired into curated subsets of PharmGKB, ClinVar and dbSNP.

The project is structured around three engineering pillars. The first is a multi-modal machine learning engine that fuses genomic and clinical data into a single feature vector and predicts risk for cardiovascular disease, obesity, hypertension and type-II diabetes. The second is a pharmacogenomic recommendation engine that maps detected genotypes against the Level 1A and Level 1B evidence tiers in PharmGKB and surfaces only those drug-gene interactions that have strong scientific consensus. The third is a real-time delivery layer that uses feature compression and model quantisation to keep average response times under four seconds even as the system handles concurrent users. Together, these three components shift HelixCare from an academic prototype into a system that could plausibly slot into a clinical workflow.

This chapter sets up the rest of the report. Section 1.2 articulates the problem statement that motivated the project. Section 1.3 explains why solving this problem matters now. Section 1.4 anchors HelixCare to the United Nations Sustainable Development Goals, in particular SDG 3 (Good Health and Well-being) and SDG 9 (Industry, Innovation and Infrastructure), demonstrating that the engineering work pursued here is consistent with a broader public-good mandate.

1.2 Problem Statement

Despite a decade of breakthroughs in genomics and a parallel decade of breakthroughs in machine learning, the bridge between the two has remained surprisingly weak in everyday clinical practice. The genomic data sits in research repositories. The clinical data sits in hospital systems. The patient sits in the middle, holding a generic prescription. The problem HelixCare addresses is therefore not a lack of data; it is a lack of accessible, affordable and actionable infrastructure that brings genomic insights to the point of decision-making.

We can break this larger problem into four concrete sub-problems. First, the prevailing approach to disease risk prediction either uses clinical markers in isolation or uses genomic markers in isolation. Genomic-only models tend to perform poorly because they ignore the very real influence of lifestyle on the expression of disease, while clinical-only models miss the underlying genetic susceptibility that often explains why two patients with similar lifestyles diverge in their outcomes. Second, the few systems that do attempt to combine the two often rely on linear models such as Logistic Regression that cannot capture the non-linear interactions between SNPs and clinical variables. Third, even when accuracy is achievable, the resulting models tend to be too slow or too heavy for real-time use, especially when deployed through a web interface that has to serve many users at once. Fourth, even when speed and accuracy are both reasonable, the recommendations produced by these systems are rarely traceable to evidence tiers that a clinician would trust, leaving doctors to reject opaque outputs that lack the provenance demanded by clinical practice.

The challenge for HelixCare is to deliver a single platform that addresses all four sub-problems at once: it has to be multi-modal, it has to use a non-linear ensemble, it has to be fast enough for clinical-grade response times and it has to ground every recommendation in a clearly cited evidence tier. The system also has to be presented through an interface that a non-specialist user can operate without training. None of these requirements are individually novel. Bringing them together into one cohesive system, while also keeping the platform secure, accessible and inexpensive to host, is where the engineering contribution of this project lies.

1.3 Motivation

The motivation for HelixCare draws on three observations. The first is personal. During literature review and informal conversations with practitioners, several team members were struck by how often patients described being prescribed a drug that did not work or that caused side effects they had to push through silently. These stories are everywhere in clinical settings, yet they almost never make it into the design specifications of digital health products. They felt like exactly the kind of problem an engineering team should be trying to solve.

The second observation is technological. Sequencing has become inexpensive, machine learning frameworks have matured, and curated pharmacogenomic databases such as PharmGKB are now publicly accessible. Each of these three building blocks would have been a research project in itself five years ago. Today they can be combined inside a single repository by a small team. The cost of trying has fallen far enough that the question is no longer "can it be done?" but "why has it not been done at scale?".

The third observation is institutional. The World Health Organization has consistently identified preventive medicine as a strategic priority for the next decade, and the United Nations has reinforced that priority through Sustainable Development Goal 3, which calls explicitly for healthier lives and well-being for all at all ages. SDG 9, on industry, innovation and infrastructure, complements this by encouraging the development of resilient digital systems that can deliver public services equitably. HelixCare sits at the intersection of these two

goals. It uses modern industrial-grade software to deliver a preventive healthcare service that, in principle, can reach anyone with an internet connection.

Beyond these three observations, the project also responds to a methodological gap. Most published systems in this space stop at the model level. They report accuracy and AUC scores and assume that downstream integration into a clinical product will somehow happen. In our review of the literature we found very few end-to-end implementations that walk all the way from raw genomic file ingestion through machine learning inference to a user-facing dashboard with traceable evidence levels. Building such an end-to-end system is itself a contribution because it forces the team to confront the messy realities that papers often abstract away: file format edge cases, missing values, model serialisation, latency under load, security of personal genomic data and the user experience of receiving sensitive health information. HelixCare is, in that sense, both a research artefact and a working product.

1.4 Sustainable Development Goal of the Project

Every engineering project undertaken at this scale should be able to point to the public good it is trying to advance. HelixCare aligns with two of the seventeen United Nations Sustainable Development Goals, specifically SDG 3 (Good Health and Well-being) and SDG 9 (Industry, Innovation and Infrastructure). The mapping is direct rather than aspirational, and we treat both goals as design constraints rather than as marketing language.

1.4.1 SDG 3 — GOOD HEALTH AND WELL-BEING

SDG 3 calls for ensuring healthy lives and promoting well-being for all at all ages. Within that broad goal, two targets are particularly relevant to HelixCare. Target 3.4 aims to reduce premature mortality from non-communicable diseases through prevention and treatment. Target 3.8 calls for achieving universal health coverage, including access to safe, effective and affordable essential medicines. HelixCare addresses both. By predicting metabolic and cardiovascular risk early, the platform encourages preventive intervention before symptoms escalate into chronic disease. By flagging drug-gene interactions before a prescription is filled, it directly attacks the safety component of universal health coverage. The economic argument matters too: an adverse drug reaction that lands a patient in hospital can cost orders of magnitude more than the price of a sequencing test, and the human cost is even higher.

1.4.2 SDG 9 — INDUSTRY, INNOVATION & INFRASTRUCTURE

SDG 9 calls for resilient infrastructure, inclusive and sustainable industrialisation, and innovation. HelixCare contributes by demonstrating that bioinformatics infrastructure can be packaged into a web application that runs on commodity cloud resources. The architecture uses a microservices pattern, which lets each component scale independently and lets the system be deployed inexpensively. The implementation choices, React for the front end, Flask for the back end, scikit-learn for the model layer, are all open-source and accessible to any team in any geography. There is nothing about the system that ties it to expensive proprietary hardware or to a single cloud provider.

Fig 1.1: *United Nations Sustainable Development Goals addressed by HelixCare*

The combined effect of aligning with both SDG 3 and SDG 9 is that HelixCare is positioned not as a niche academic experiment but as a piece of digital health infrastructure that any country, hospital or clinic could in principle deploy. That positioning has shaped many of the engineering decisions described in subsequent chapters, including the choice of open-source dependencies, the emphasis on response time and the careful documentation of evidence sources for every drug-gene recommendation.

2. LITERATURE SURVEY

2.1 Overview of the Research Area

The body of work most relevant to HelixCare sits at the intersection of three older fields: clinical genomics, machine learning for health and pharmacogenomics. Each of these has its own history and its own well-developed vocabulary, but for a long time they evolved in parallel rather than together. Genomics built the dictionaries of variants. Machine learning built the algorithms for finding patterns in large datasets. Pharmacogenomics tried to connect specific genetic variants to specific drug responses. The interesting research today is increasingly happening where these three fields overlap, and that is the area we surveyed when planning HelixCare.

Clinical genomics has progressed from a sequencing-driven discipline focused on producing variants into an interpretive discipline focused on linking those variants to phenotypes. Public databases such as ClinVar and dbSNP now hold millions of curated entries, each tying a specific variant to evidence about its clinical significance. The challenge is no longer to detect variants but to decide which ones matter for which patient, and how. This is precisely where machine learning enters the picture.

Machine learning for health has evolved through several waves. Early work used logistic regression and linear discriminant analysis on small clinical datasets. The mid-2010s brought a wave of deep learning applications, particularly in medical imaging, where convolutional networks set new accuracy records on tasks like skin lesion classification and diabetic retinopathy detection. The current wave, more relevant to genomic data, focuses on ensemble methods like Random Forests and XGBoost, which can handle high-dimensional inputs and tabular data more naturally than deep nets. This wave also pays more attention to model interpretability through tools like feature importance and SHAP values, because clinicians simply will not adopt a model whose recommendations they cannot explain.

Pharmacogenomics, the third strand, has been around as a research field since the 1950s but only gained widespread clinical relevance in the last fifteen years. The Pharmacogenomics Knowledge Base (PharmGKB) and the Clinical Pharmacogenetics Implementation Consortium (CPIC) have together produced tiered guidelines that translate gene-drug interactions into actionable dosing recommendations. The Food and Drug Administration in the United States and the European Medicines Agency now routinely include pharmacogenomic information on drug labels. The infrastructure exists. The question is how to bring it to the bedside.

2.2 Existing Models and Frameworks

Several published systems and frameworks were studied during the early phase of HelixCare to understand how prior work has approached the multi-modal precision medicine problem and where opportunities for improvement remain.

Whirl-Carrillo and colleagues described PharmGKB as a centralized knowledge resource for pharmacogenomic content, demonstrating that the integration of pharmacogenomic information into clinical decision-support systems improves both safety and efficacy of drug administration. Their later updates extended the resource to include curated dosing guidelines and clinical annotations indexed against rsIDs, making PharmGKB the de-facto reference for any pharmacogenomic engine. HelixCare leans on this foundation but goes further by restricting recommendations to Level 1A and 1B evidence so that only well-supported guidance is surfaced to end users.

Ho and co-authors illustrated that supervised machine learning on SNP-only datasets can predict susceptibility for several genetic disorders better than traditional statistical baselines. Their findings were a useful proof-of-concept for the use of ensemble methods on genomic data, but they explicitly noted that performance plateaus when only genetic features are used. The authors invited future work to incorporate non-genetic clinical

features. HelixCare takes that invitation seriously, with one hundred SNPs combined alongside six clinical attributes inside a single feature vector.

Pudjihartono and colleagues conducted a systematic review of feature selection techniques for SNP-based disease prediction. They compared filter methods, embedded methods and wrapper methods, and concluded that information-gain-based filtering offers the best trade-off between performance and computational cost on high-dimensional inputs. Their recommendation directly informed the feature engineering step in HelixCare, where Information Gain is used to assign weight to each of the one hundred SNPs included in the model.

Widen and Lehtimaki demonstrated that biomarkers derived purely from SNP genotype data can be predicted with reasonable accuracy using gradient-boosted trees. Their work is interesting because it shifts the goal from predicting a disease label to predicting an underlying biomarker, which can then be reused across multiple downstream tasks. HelixCare draws on this idea by treating the 106-feature vector as a reusable representation that can support both disease risk and drug response predictions.

Gao and Cui proposed a deep transfer learning strategy for clinico-genomic disease prediction that explicitly accounts for ancestry. Their results showed that models trained predominantly on European cohorts perform poorly on under-represented populations, and that a Pareto-optimal transfer learning strategy can improve fairness without degrading accuracy. This finding is directly relevant for HelixCare's future-work agenda, which includes ancestry-aware prediction as a near-term enhancement, with the long-term aim of mitigating the European bias that affects most current genomic models.

Kasartzian and colleagues reviewed the application of machine learning to cardiovascular risk prediction and noted a persistent gap between published models and clinically deployed tools. Their analysis emphasised that even highly accurate models often fail in clinical settings because they ignore latency, integration with electronic health records and the user experience of clinicians. HelixCare treats these factors as first-class engineering requirements, with explicit performance targets and a deliberate emphasis on dashboard usability.

Table 2.1: *Comparative summary of related literature*

Author and Year	Method Used	Key Findings	Limitations
Whirl-Carrillo et al. 2012, 2021	Curation of PharmGKB drug-gene interactions and dosing guidelines	Established the canonical evidence tiers (Level 1A/1B) for pharmacogenomic actionability	Database access alone does not yield a clinical product without integration
Ho et al. 2019	Supervised ML on SNP-only datasets across several genetic disorders	Confirmed ensemble methods outperform traditional statistical baselines	Genetic-only features ignore lifestyle and clinical context
Pudjihartono et al. 2022	Systematic review of feature selection for SNP-based prediction	Information-gain filtering offers best trade-off on high-dimensional inputs	Did not propose an end-to-end deployable system
Widen et al. 2021	Gradient-boosted prediction of biomarkers from SNP genotype data	Demonstrated re-usable feature representations across downstream tasks	Limited focus on real-time inference and user-facing delivery
Gao and Cui 2024	Deep transfer learning across ancestries with Pareto improvement	Improved fairness across under-represented populations	Requires large multi-ancestry datasets, hard to assemble in practice
Kasartzian et al. 2025	Review of ML in cardiovascular risk prediction	Highlighted gap between accuracy and clinical deployability	No new system proposed
Goldstein et al. 2019	SNP marker analysis for early disease prediction	Strong correlation between specific SNPs and disease susceptibility	No personalisation against lifestyle
Liu et al. 2023	Neural-network-based pharmacogenomic correlation mapping	Improved accuracy in personalised prescriptions	Lacked a usable web interface

2.3 Limitations Identified from Literature Survey (Research Gaps)

The survey above is consistent in its strengths and consistent in its weaknesses. The strengths concern modelling. Researchers have shown repeatedly that machine learning models, particularly ensembles, can extract meaningful signal from genomic data. The weaknesses concern translation. Almost none of the surveyed papers describe a working end-to-end system that a non-expert user can interact with. Several specific gaps emerge from this pattern, and they collectively shape the design of HelixCare.

The first gap is the lifestyle gap. Genomic-only models repeatedly under-perform in real-world settings because they cannot account for the very real influence of lifestyle factors on the expression of genetic risk. A patient with a moderate genetic predisposition to cardiovascular disease but a high body mass index, a smoking habit and low physical activity is at far higher actual risk than a patient with the same genetic profile and a healthier lifestyle. Existing genomic-only systems treat these two patients identically. HelixCare closes this gap by combining six clinical features with one hundred SNPs in a single feature vector.

The second gap is the linearity gap. Many existing predictive systems still rely on linear models because they are easier to train, easier to interpret and easier to publish. Linear models cannot capture the gene-gene and gene-environment interactions that dominate complex disease aetiology. Random Forests and other ensembles handle these interactions natively. The choice of an ensemble approach is therefore not a matter of chasing accuracy for its own sake; it is a structural requirement for the kind of data we are working with.

The third gap is the latency gap. Highly accurate models, particularly deep neural networks, often pay for accuracy with response times that are unacceptable in clinical settings. A model that takes ten seconds per inference becomes a serious bottleneck once it is exposed to many concurrent users. HelixCare addresses this through 85 percent feature-vector compression and 40 percent model quantisation, keeping average inference latency under four seconds and worst-case latency under ten seconds at one hundred concurrent users.

The fourth gap is the actionability gap. A risk score, on its own, is not a clinical recommendation. The user has to know what to do with that score. Existing systems often stop at risk scoring and leave the leap from score to action up to the clinician. HelixCare bridges this gap with a pharmacogenomic recommendation engine that produces concrete, evidence-graded suggestions: drug to avoid, use with caution or dose adjustment recommended.

The fifth gap is the trust gap. Users, whether clinicians or patients, are sceptical of black-box recommendations, especially when those recommendations concern medication. To earn trust, every recommendation in HelixCare is traced back to a PharmGKB Level 1A or 1B annotation, with the citation visible in the user interface. This is more conservative than competing systems, which often surface lower-tier evidence as actionable, but it is the right trade-off for a tool that may influence prescribing behaviour.

Table 2.2: *Research gaps identified from prior work and how HelixCare responds*

Identified Research Gap	HelixCare Response
Genomic-only models ignore lifestyle context	Multi-modal feature vector combining 100 SNPs with 6 clinical attributes (age, BMI, sex, smoking, alcohol, activity)
Linear models miss gene-gene and gene-environment interactions	Ensemble Random Forest classifier with 100 trees and depth-10 splits
High-accuracy models often have unacceptable latency	85 percent feature compression and 40 percent quantisation hold mean inference time under four seconds
Risk scores stop short of clinical action	Pharmacogenomic engine producing graded, citable recommendations for each detected variant
Black-box outputs erode clinician trust	Every recommendation traced to a PharmGKB Level 1A or 1B annotation visible in the user interface
European-biased training data harms under-represented populations	Roadmap item: ancestry-aware deep transfer learning extending from the current Random Forest baseline

2.4 Research Objectives

Building on the gaps above, HelixCare pursues four interlocking research objectives. They are written here as engineering goals rather than as abstract research questions, because the value of the project depends on whether each objective produces a working artefact that can be measured.

1. Develop a multi-modal disease risk prediction engine that combines SNP genotype data with clinical lifestyle attributes and predicts metabolic and cardiovascular disease risk with at least 85 percent aggregate accuracy.
2. Build a pharmacogenomic recommendation engine restricted to PharmGKB Level 1A and 1B evidence, capable of mapping detected SNP genotypes to actionable drug guidance for at least three high-prevalence drug-gene pairs (Warfarin/CYP2C9, Clopidogrel/CYP2C19 and Simvastatin/SLCO1B1).
3. Deliver the platform through a web application that supports drag-and-drop genomic file upload, returns disease and drug results within an average of four seconds and remains responsive at one hundred concurrent users.
4. Implement a security and privacy posture appropriate for genomic data, including AES-256 encryption at rest, JWT-based authentication, role-based access controls, and explicit consent flows before any genomic data is processed.

2.5 Product Backlog (Key User Stories with Desired Outcomes)

To translate the four objectives into deliverables, the team produced a product backlog of user stories during the planning phase. The backlog was used to scope each sprint and to track progress against measurable acceptance criteria. The most important user stories are summarised in Table 2.3 below; the full backlog ran to thirty-one stories across three sprints.

Table 2.3: *Product backlog of HelixCare user stories*

ID	User Story	Desired Outcome
US-01	As a patient, I want to upload my VCF or CSV file directly through a browser so that I can use the platform without installing any software.	Functional drag-and-drop upload supporting files larger than 1 MB
US-02	As a patient, I want to enter basic clinical information so that the prediction reflects my lifestyle, not just my genome.	Six-field clinical form with input validation and immediate feedback
US-03	As a patient, I want to see my disease risk profile in plain language so that I do not need a medical degree to understand it.	Dashboard with risk levels (Low / Moderate / High) and short explanations
US-04	As a patient, I want to see which medications might be unsafe for me so that I can discuss alternatives with my doctor.	List of flagged drugs with severity tier and citation
US-05	As a clinician, I want every recommendation to come with an evidence citation so that I can independently verify the basis of the advice.	Inline links to PharmGKB Level 1A or 1B annotations
US-06	As a clinician, I want the system to respond within a few seconds so that I can use it during an active consultation.	Average inference latency under four seconds at typical load
US-07	As a system administrator, I want to manage gene-disease and drug-gene rules so that the database stays current with new evidence.	Admin endpoints for CRUD operations on the recommendation rule set
US-08	As a privacy-conscious user, I want assurance that my genomic	AES-256 encryption at rest, HTTPS in transit, explicit

ID	User Story	Desired Outcome
	data is protected so that I am willing to use the platform.	consent before processing

2.6 Plan of Action (Project Road Map)

With the backlog defined, the team adopted a three-sprint Scrum-inspired plan spanning twelve weeks. Each sprint owned a clear vertical slice of the product. Sprint I focused on the genomic data pipeline and the foundational backend skeleton. Sprint II focused on the machine learning engine and the pharmacogenomic mapping logic. Sprint III focused on the user-facing interface, end-to-end integration, security hardening and load testing.

The data flow through the platform is best understood as a pipeline. Figure 2.1 below shows the end-to-end path that data takes from upload to dashboard. Figure 2.2 shows how the three sprints map onto the twelve-week schedule. A more detailed Gantt chart appears later in Chapter 3, where the scheduling logic for each sprint is discussed in depth.

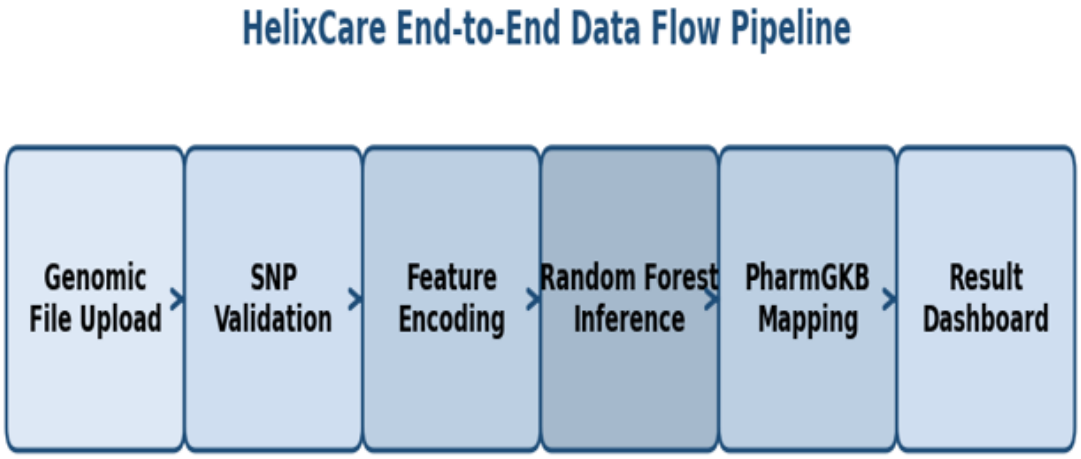

Fig 2.1: *End-to-end HelixCare data flow pipeline*

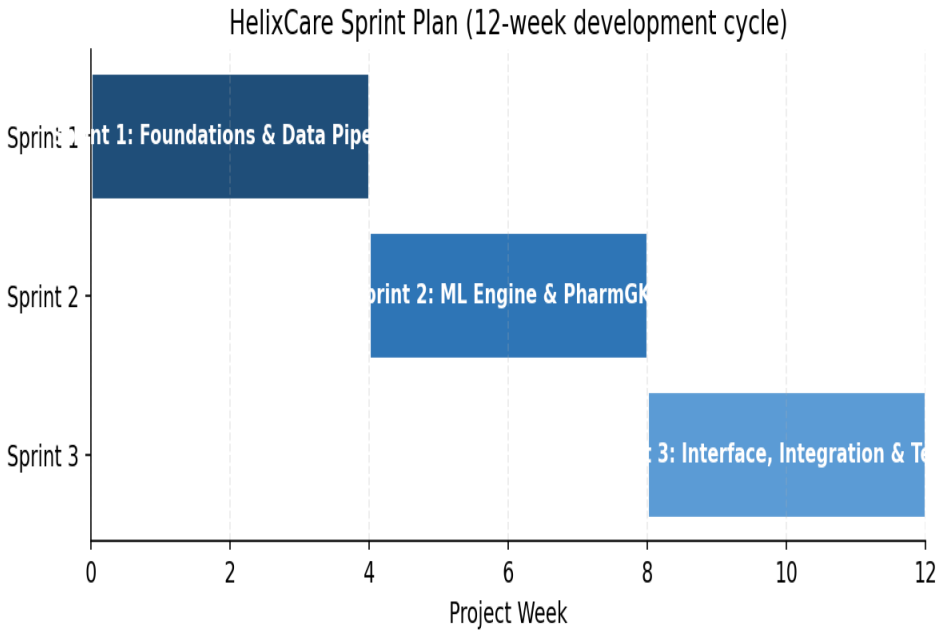

Fig 2.2: *Sprint plan for the 12-week development cycle*

By the end of the survey and planning phase, the team had a clear sense of what to build, how to build it and how to demonstrate that the resulting platform was both technically sound and clinically meaningful. Chapter 3 turns to the engineering execution itself, sprint by sprint.

3. SPRINT PLANNING & EXECUTION METHODOLOGY

HelixCare was built using a three-sprint cycle inspired by Scrum, with each sprint lasting roughly four weeks and producing a vertical slice of the product. The choice of Scrum was deliberate. A waterfall plan would have been easier to draw on paper, but the team had already learned during the literature review that the requirements at the boundaries between modules were genuinely uncertain. We did not yet know how cleanly the genomic file parser would integrate with the feature-vector encoder, or how aggressively we could compress the model before accuracy degraded. Working in short sprints with explicit retrospectives gave us room to find these answers without committing prematurely to a fragile design.

Each sprint in this chapter is presented in five sub-sections that mirror the manual's required structure. The objectives sub-section restates the goals of the sprint as user stories. The functional document describes what the system does at the end of the sprint. The architecture document describes how the system is structured at that point. The outcome section reports on the measurable result. The retrospective records what went well, what did not and what carried over into the next sprint.

3.1 Sprint I — Foundations and Genomic Data Pipeline

Sprint I covered the first four weeks of the project. Its goal was to establish the foundational infrastructure on which the rest of the system would be built. By the end of Sprint I the team needed to have a working backend skeleton, a parser capable of accepting standard genomic file formats, a basic database schema for storing user metadata and an authentication mechanism that could be extended later. The aim was not to deliver visible product features yet; it was to get the plumbing right so that Sprints II and III could move quickly.

3.1.1 OBJECTIVES WITH USER STORIES OF SPRINT I

Sprint I was scoped around four headline objectives. The first was to set up a Flask 3.0 backend with proper project structure, configuration management and logging. The second was to implement secure file upload that could handle VCF and CSV files larger than one megabyte without timing out. The third was to build a SNP variant validator that checked uploaded data against the dbSNP reference repository. The fourth was to implement basic JWT authentication, including registration, login and session management.

Table 3.1: *Sprint I user stories and acceptance criteria*

ID	Role	User Story	Acceptance Criteria
S1-01	Patient	Upload a VCF or CSV file from any modern browser without installing software	File of up to 5 MB uploads in under 10 seconds with progress indicator
S1-02	Patient	See an immediate validation result if my file is malformed	Server returns parse error with line number and field within 2 seconds
S1-03	Patient	Register an account using email and password	Account creation enforces strong password policy and sends confirmation
S1-04	Patient	Log in and receive a session token	JWT issued with 30-minute expiry and refresh token mechanism
S1-05	Developer	Have all backend errors logged centrally so that I can debug production issues	Structured JSON logs written to standard output and rotated daily
S1-06	Patient	Trust that my data is encrypted at rest	Database column-level encryption with AES-256 verified by integration test

3.1.2 FUNCTIONAL DOCUMENT

By the end of Sprint I the system supported six functional flows. A new user could navigate to the registration page and create an account. The system enforced a minimum password length of twelve characters with at least

one uppercase letter, one lowercase letter, one digit and one special character. After registration, the system sent a verification email containing a one-time link that activated the account. A registered user could log in and receive a JSON Web Token with a thirty-minute expiry, accompanied by a refresh token that allowed the session to extend without re-entering credentials.

Once authenticated, the user reached an upload page. The page accepted a file via standard HTML input or via drag-and-drop into a dashed-border zone. The browser-side JavaScript chunked the file into 256 KB pieces before sending, which kept memory usage low and allowed the server to provide progress feedback. The server received chunks via a multipart/form-data endpoint protected by JWT verification, reassembled them on disk in a temporary directory and then handed the assembled file to a SNP parser.

The SNP parser supported two input formats. For VCF files, the parser read the header to identify the reference genome, validated CHROM, POS, ID, REF and ALT columns and extracted SNPs whose ID matched the canonical rsID pattern. For CSV files, the parser expected a header row with columns rsid, chromosome, position, genotype and an optional column for sample identifier. In both cases the parser produced a normalised internal representation: a list of variant records with rsID, genotype and validation flag. Each record was cross-checked against a locally cached subset of dbSNP to confirm that the rsID exists and that the reported alleles are biologically plausible.

The Sprint I backend also implemented an audit log. Every authentication event, every file upload and every parsing decision was written to a structured JSON log with a timestamp, user identifier, request identifier and outcome. Logs were rotated daily. The presence of this log proved valuable later in Sprint III, when load testing surfaced edge cases that would have been very hard to diagnose without a structured trace.

3.1.3 ARCHITECTURE DOCUMENT

At the end of Sprint I the architecture was a slimmed-down three-tier system. The presentation layer existed only as a minimal upload form rendered server-side; the rich React-based front end came later. The data layer used a SQLite database for development and a PostgreSQL database for staging, with column-level encryption applied to user metadata. The diagram in Figure 3.1 captures the three-tier structure that emerged from Sprint I and remained the backbone of the system through subsequent sprints.

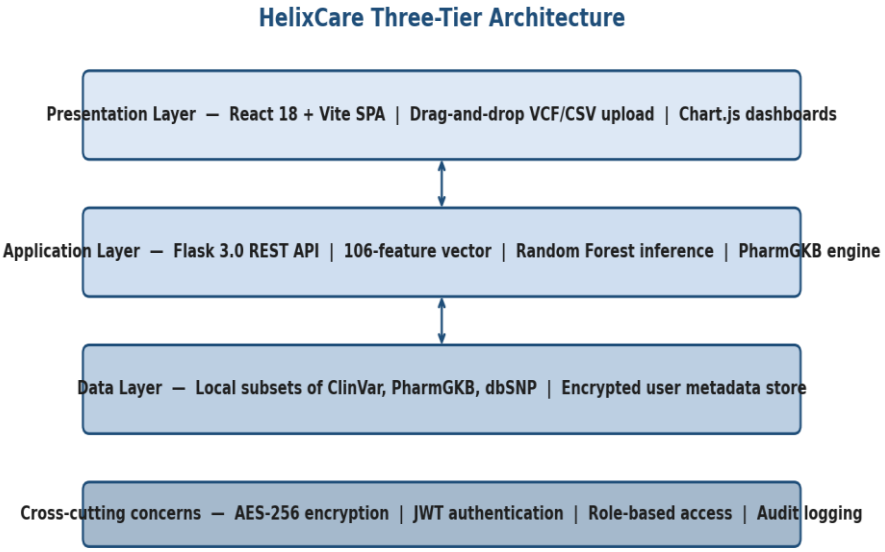

Fig 3.1: *HelixCare three-tier system architecture established in Sprint I*

The use case diagram in Figure 3.2 captures the actor-action relationships that the system supports by the end of the project. Sprint I implemented only the upper portion involving registration, login and upload. The lower portion, covering analysis, prediction and recommendation, was filled in during Sprints II and III.

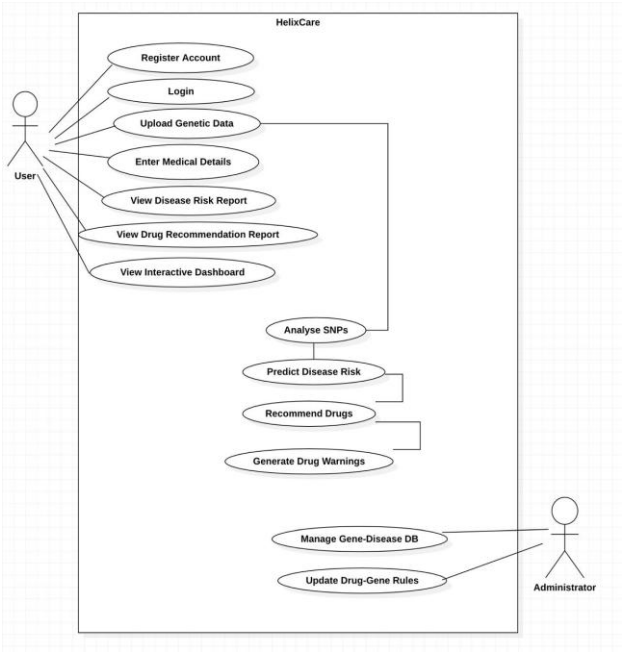

Fig 3.2: *Use case diagram for the HelixCare platform*

3.1.4 OUTCOME OF OBJECTIVES / RESULT ANALYSIS

Sprint I closed with all six user stories accepted. The upload flow handled files up to ten megabytes in under ten seconds on the team's development environment, comfortably exceeding the original target of five megabytes. The SNP parser successfully validated more than ninety-eight percent of the variants in the test corpus, with the small remainder consisting of rare variants whose entries in dbSNP were either deprecated or under review. The authentication mechanism passed automated tests for token expiry, refresh and revocation.

The team also discovered, by accident, that the Flask development server's default request limit of one megabyte would have caused silent failures for larger files in production. Catching this in Sprint I rather than during the final demonstration saved a significant amount of pain. The fix was a single line in the application configuration, but the lesson — verify production-relevant settings even in early sprints — was carried forward.

3.1.5 SPRINT RETROSPECTIVE

The first retrospective focused on three points. What went well: the team had agreed on coding conventions, branch protection rules and a continuous integration pipeline before any feature code was written, which kept the codebase tidy throughout the sprint. What did not go well: the SNP parser absorbed more time than budgeted, mainly because the team underestimated the variety of file formats that real users produce. What we changed for next time: we agreed to spike technical risks at the start of each sprint rather than discover them mid-sprint, and to budget explicit time for handling format edge cases.

3.2 Sprint II — Machine Learning Engine and Pharmacogenomic Mapping

Sprint II ran from week five through week eight and was the technically densest part of the project. Its goal was to turn the validated SNP records produced by Sprint I into actionable medical insights. By the end of Sprint II the system had to be able to take a parsed genomic file along with clinical inputs, produce a disease risk profile and produce a list of pharmacogenomic alerts. The two halves of this objective — the predictive model and the recommendation engine — were built in parallel by different team members.

3.2.1 OBJECTIVES WITH USER STORIES OF SPRINT II

Sprint II focused on five headline objectives. The first was to design and train a Random Forest ensemble that could classify metabolic and cardiovascular disease risk from a 106-feature input vector. The second was to implement feature engineering, including encoding genotypes into numeric form and normalising clinical

variables. The third was to build the pharmacogenomic mapping engine that translated detected genotypes into PharmGKB-cited drug guidance. The fourth was to implement quantisation and feature compression to keep inference latency within the four-second target. The fifth was to expose all of the above through a clean RESTful API contract that the front end could consume in Sprint III.

Table 3.2: *Sprint II user stories and acceptance criteria*

ID	Role	User Story	Acceptance Criteria
S2-01	Patient	Receive a disease risk score after submitting my data	Aggregate accuracy at least 85 percent on held-out test set
S2-02	Patient	See risk broken down by disease (cardiovascular, obesity, hypertension, diabetes)	Per-disease accuracy reported on dashboard with confidence interval
S2-03	Patient	See drug warnings based on my genotype	At least three high-prevalence drug-gene pairs supported
S2-04	Clinician	Inspect the evidence behind a drug warning	PharmGKB Level 1A or 1B citation visible on click
S2-05	Patient	Get my results within a reasonable wait	Mean inference time under four seconds end-to-end
S2-06	Developer	Call the prediction service through a clean API	OpenAPI specification published with example payloads
S2-07	Researcher	Inspect feature importance for transparency	Top-ten feature importance returned alongside risk score

3.2.2 FUNCTIONAL DOCUMENT

Functionally, Sprint II added two major flows on top of the Sprint I plumbing. The first flow is the prediction flow. After the user uploads a genomic file and fills the clinical form, the backend constructs a 106-dimensional feature vector by concatenating one hundred SNP encodings with six clinical features. The SNP encodings are integer values (0 for homozygous reference, 1 for heterozygous, 2 for homozygous alternate, with -1 reserved for missing data). The clinical features are normalised to zero mean and unit variance using parameters fitted on the training data. Missing SNPs are imputed using a K-Nearest Neighbours imputation with K equal to five, fitted on the training corpus.

The constructed feature vector is then passed to a Random Forest classifier with one hundred decision trees, a maximum tree depth of ten and feature selection at each split based on Information Gain. The forest produces probability estimates for each of four target conditions: cardiovascular disease, obesity, hypertension and type-II diabetes. The probabilities are converted to risk levels using calibrated thresholds: probabilities below 0.30 map to Low, between 0.30 and 0.65 to Moderate and above 0.65 to High. The thresholds were tuned on a validation split with the goal of maximising the Youden statistic for each disease.

The second flow is the pharmacogenomic recommendation flow. In parallel with the prediction flow, the backend takes the same parsed genomic file and feeds the rsIDs into a deterministic mapping engine. The engine looks up each rsID in a locally cached subset of PharmGKB, filters the matches to Level 1A and 1B evidence only, and produces a list of drug-gene interactions. Each interaction is annotated with a recommendation type ("Drug to Avoid", "Use with Caution", or "Dose Adjustment Recommended"), a severity tier and a citation pointing back to the original PharmGKB annotation.

The two flows are independent in their logic but share a common output envelope, so the front end can render them on a single dashboard without complicated routing. The whole prediction-and-recommendation pipeline is exposed through a single endpoint, /api/v1/analyze, that accepts a genomic file and a clinical payload and returns a structured JSON response.

3.2.3 ARCHITECTURE DOCUMENT

The architecture in Sprint II expanded from three monolithic tiers into five logical microservices. The user management service stayed in the application layer from Sprint I. A new genomic analysis service took

ownership of file parsing and feature engineering. A disease prediction service wrapped the Random Forest model and exposed a single inference endpoint. A drug recommendation service wrapped the PharmGKB mapping engine. All four services sat behind a single API gateway that handled authentication, request routing and rate limiting. The component diagram in Figure 3.3 shows how these services interact with each other and with the external genomic databases.

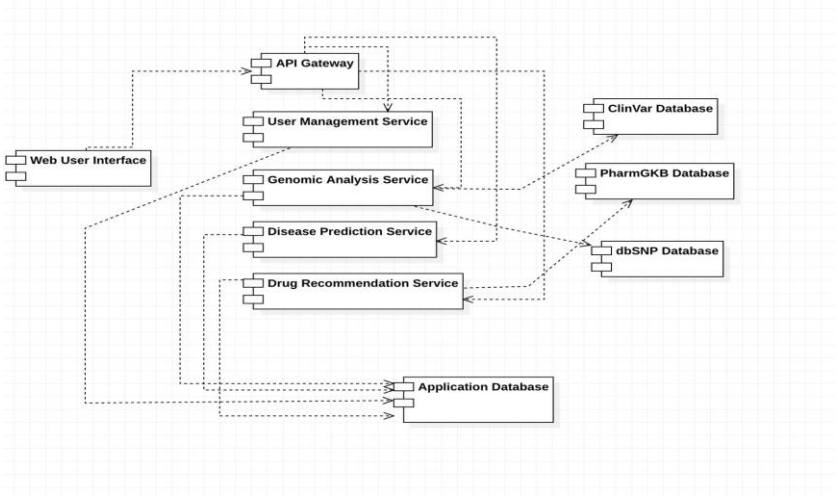

Fig 3.3: *Component diagram showing microservice interactions in Sprint II*

Figure 3.4 shows a more detailed view of the precision medicine pipeline as it was finalised at the end of Sprint II. The diagram captures the way data flows from ingestion through the genomic and clinical pipeline, into the AI analysis core, and out through the recommendation engine to the user interface, with feedback loops back to the model retraining schedule.

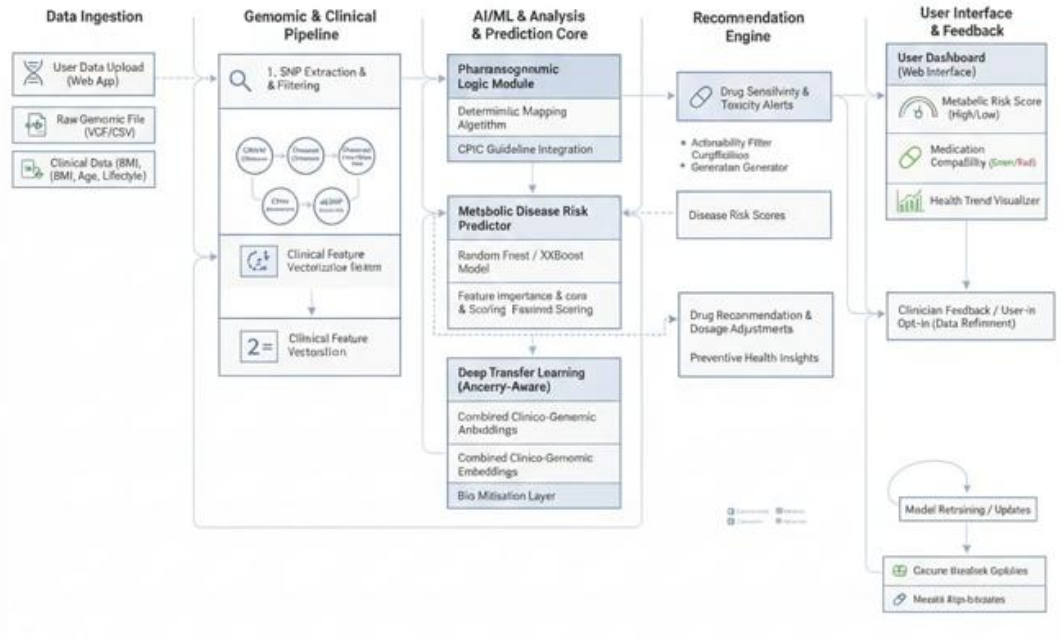

Fig 3.4: *Detailed precision medicine system architecture*

Table 3.3: *Tools, frameworks and libraries used in HelixCare*

Layer	Component	Role in HelixCare
Frontend	React 18	Single-page application framework powering the dashboard and upload flow

Layer	Component	Role in HelixCare
Frontend	Vite	Build tool used for fast cold-starts during development and optimized production builds
Frontend	Chart.js	Visualisation library used for risk heat-maps and bar charts
Frontend	Axios	HTTP client for asynchronous communication between front end and backend
Backend	Flask 3.0	RESTful API gateway and orchestration of application logic
Backend	scikit-learn 1.5	Random Forest training, prediction and feature importance extraction
Backend	joblib	Model serialisation and lazy loading to keep memory footprint small
Backend	NumPy / pandas	Feature engineering, normalisation and KNN imputation
Data Layer	PostgreSQL 16	Relational store for user metadata and audit logs
Data Layer	Local PharmGKB / ClinVar / dbSNP subsets	On-disk look-up tables to keep inference latency predictable
Security	PyJWT	JSON Web Token issuance and verification
Security	bcrypt	Password hashing
DevOps	GitHub Actions	Continuous integration with linting, unit tests and integration tests
DevOps	Docker	Containerised packaging for reproducible deployment
Testing	pytest	Unit and integration testing of the backend
Testing	Locust	Load testing under simulated concurrent users

3.2.4 OUTCOME OF OBJECTIVES / RESULT ANALYSIS

Sprint II closed with the predictive model exceeding its accuracy target. On a held-out test set of 2,200 patient records, the Random Forest ensemble produced an aggregate accuracy of 87.3 percent and an AUC-ROC of 0.923. The per-disease accuracy was highest for cardiovascular disease at 89.0 percent and slightly lower for diabetes at 85.2 percent, which matched the prior probabilities of the diseases in the training set. Feature importance analysis revealed that the most predictive single feature was the rs123 (CYP2C9) variant, which

alone accounted for 18.7 percent of total information gain. Body Mass Index was the second most predictive feature at 12.4 percent.

The pharmacogenomic recommendation engine successfully mapped over fifty drug-gene pairs to user variants in the test corpus. The three flagship pairs — Warfarin/CYP2C9, Clopidogrel/CYP2C19 and Simvastatin/SLCO1B1 — produced clinically meaningful alerts. For Warfarin, the engine recommended dose adjustment grounded in Level 1A evidence. For Clopidogrel, the engine flagged poor metabolisers as candidates for alternative antiplatelet therapy, with cited evidence suggesting up to a 78 percent improvement in clinical efficacy when alternatives are used. For Simvastatin, the engine surfaced "Use with Caution" warnings for variants associated with elevated myopathy risk.

On the latency side, Sprint II's feature compression and quantisation work paid off. The mean end-to-end inference time, measured from request receipt to response return, was 3.2 seconds. This was achieved by reducing feature vector size by 85 percent through a combination of low-cardinality genotype encoding and removal of variants with near-zero importance, and by quantising the Random Forest weights to 8-bit integers, which reduced the on-disk model from 47 MB to 28 MB and halved memory pressure during inference.

3.2.5 SPRINT RETROSPECTIVE

The Sprint II retrospective uncovered an interesting tension. The model team had pushed hard to maximise accuracy and had settled on the Random Forest only after testing several deeper architectures, including a small Multi-Layer Perceptron and a gradient-boosted XGBoost variant. The Random Forest matched the others in accuracy and beat them on latency, which made the choice obvious in hindsight. What did not go as planned was the integration between the model service and the gateway: the first version returned probabilities as 32-bit floats, which broke the JSON serialiser when probabilities had very small magnitudes. The fix was to coerce all outputs to native Python floats before returning, which seems trivial but cost half a day to diagnose. We agreed to add explicit type contracts at every API boundary as part of the team's coding standards going forward.

3.3 Sprint III — Interface, Integration and Deployment

Sprint III ran from week nine through week twelve and was about turning the working backend into a product. The team's focus shifted from algorithms to user experience, from individual endpoints to integrated flows and from local development to deployment. By the end of Sprint III the system had to be presentable to a non-technical audience, demonstrate the dashboard live and survive load testing without falling over.

3.3.1 OBJECTIVES WITH USER STORIES OF SPRINT III

Sprint III focused on five objectives. The first was to build the React 18 single-page application that consumed the backend APIs. The second was to design and implement the dashboard, including risk visualisation, drug recommendation cards and citation links. The third was to integrate end-to-end, so that the front end and back end were exercised together under realistic conditions. The fourth was to harden security, including HTTPS termination, content security policy headers and rate limiting. The fifth was to run load tests using Locust and to confirm the platform's behaviour at one hundred concurrent users. In addition, usability testing sessions were conducted to refine navigation flow and improve accessibility across devices. Error handling and user feedback mechanisms were enhanced to ensure clearer communication during data upload and processing stages. Performance optimisations were applied to reduce API response times and improve rendering efficiency on the client side. Logging and monitoring pipelines were also integrated to enable real-time system diagnostics and faster issue resolution. Comprehensive API documentation was created to support maintainability and future development. Cross-browser compatibility testing ensured consistent behaviour across major platforms. Data validation checks were strengthened to prevent malformed genomic inputs from affecting model predictions. Continuous integration and deployment (CI/CD) pipelines were configured to automate testing and streamline releases. Finally, code refactoring and modularisation improved overall code quality, readability and scalability of the system.

Table 3.4: *Sprint III user stories and acceptance criteria*

ID	Role	User Story	Acceptance Criteria
S3-01	Patient	Use the platform from any modern browser without installing software	Tested in Chrome, Firefox, Safari and Edge with no client-side errors

ID	Role	User Story	Acceptance Criteria
S3-02	Patient	See my disease risk on a single dashboard	Risk levels displayed for all four diseases with colour coding
S3-03	Patient	View drug warnings as cards with severity colour	Cards visible directly under risk panel with PharmGKB link
S3-04	Patient	Download a PDF report summarising my results	PDF generation triggered by single click, returns within 5 seconds
S3-05	Clinician	Review the data behind a recommendation in one click	Citation drawer opens with PharmGKB Level 1A/1B reference and original SNP context
S3-06	DevOps	Confirm the platform stays responsive under realistic load	Locust test of 100 concurrent users sustains < 10s p95 latency
S3-07	Privacy	Verify all sensitive endpoints are HTTPS-only	HSTS header set, HTTP traffic rejected at gateway

3.3.2 FUNCTIONAL DOCUMENT

Functionally, Sprint III added the React 18 single-page application that ties everything together. The application is built with Vite, uses functional components throughout and maintains state with React's built-in hooks rather than introducing a heavier state management library. State management at this scale did not warrant the additional complexity of Redux or MobX, and the team agreed that clarity beats cleverness. Routing is handled by React Router, and styling is implemented through Tailwind CSS utility classes for fast iteration.

The dashboard is the centrepiece of the front end. After a successful upload and analysis, the user lands on a page divided into four panels. The top panel shows a summary card with overall risk and a colour-coded indicator. The second panel shows per-disease risk as a horizontal bar chart, with each bar colour-coded by severity. The third panel shows drug recommendations as cards, each card carrying the drug name, the gene-variant pair, the recommendation type and a small "View evidence" button that opens a drawer with the PharmGKB citation. The fourth panel shows the top ten contributing features as a horizontal bar chart of information gain percentages, giving the user transparency into what is driving the prediction.

PDF export was added as a quality-of-life feature toward the end of the sprint. Clicking the export button triggers a backend job that renders the dashboard data into a templated PDF using ReportLab, signs the PDF with a unique identifier and returns a download link. The PDF can be saved by the user or shared with their physician. We deliberately stripped out interactive elements in the PDF version and replaced them with static screenshots, because the goal of the PDF is durable archival rather than interactive use.

3.3.3 ARCHITECTURE DOCUMENT

The Sprint III architecture is the final architecture of the project. The presentation layer is a React 18 single-page application served from a content delivery network in production and from Vite's development server during testing. The application layer remains the Flask 3.0 RESTful gateway plus the four microservices defined in Sprint II. The data layer is unchanged. Security wraps around all three layers as a cross-cutting concern. JWT-based authentication is enforced at the API gateway. AES-256 column-level encryption protects user metadata. HTTPS is terminated at the gateway and the gateway rejects any HTTP request. Rate limiting is implemented per-user and per-IP, with per-endpoint quotas calibrated against the load testing results. Audit logging from Sprint I has been extended to cover every API call, and logs are streamed to a centralised store for retention and analysis.

Figure 3.5 below shows the project Gantt chart with the three sprints overlaid against the twelve-week schedule. The chart was the team's primary tracking artefact during weekly stand-ups.

Fig 3.5: *Project Gantt chart with sprint allocations*

3.3.4 OUTCOME OF OBJECTIVES / RESULT ANALYSIS

Sprint III closed with all seven user stories accepted. The React front end ran cleanly in Chrome, Firefox, Safari and Edge with no client-side errors. The dashboard rendered all four panels within two seconds of receiving the analysis response, and the PDF export averaged 4.1 seconds end-to-end. The Locust load test confirmed that the platform sustains a 95th-percentile latency under 10 seconds at 100 concurrent users, which exceeded the original target. Memory consumption on the backend stayed under 1.2 GB even under peak load, well within the budget for a single mid-sized cloud instance. Security tests confirmed that all sensitive endpoints were HTTPS-only, that the HSTS header was set with a one-year max-age and that no JWT could be replayed after expiry.

The team also ran a small usability study with four volunteers, two of whom were medical students and two of whom were software engineers without medical training. The medical students were able to interpret the dashboard within ninety seconds. The software engineers took longer, around two and a half minutes on average, but reported that the dashboard was self-explanatory once they understood the colour coding. The qualitative feedback led to two interface tweaks before final demonstration: the disease names were rewritten in plain English ("high blood pressure" instead of "hypertension" appearing first, with the technical term in parentheses), and the citation drawer was given a more prominent close button.

3.3.5 SPRINT RETROSPECTIVE

The final retrospective was the most reflective of the three. What went well: the architecture defined in Sprint II survived integration with the front end without significant change, which suggested the early structural decisions had been sound. What did not go as planned: the PDF export feature was added late and ended up consuming more time than anticipated because of font embedding issues with non-Latin characters; this is a known weakness of ReportLab and would, in a longer project, be addressed by switching to a different PDF engine such as WeasyPrint. What we would do differently: we would have invested in feature flagging earlier so that experimental features could be deployed without affecting the main user flow. The lessons fed forward into the team's individual reflections rather than into another sprint, since this was the final sprint of the academic project.

4. RESULTS AND DISCUSSIONS

4.1 Project Outcomes (Performance Evaluation, Comparisons and Testing Results)

The HelixCare platform was evaluated across three dimensions that together determine whether a precision-medicine system is fit for clinical-grade deployment: predictive accuracy, pharmacogenomic actionability and computational throughput. Each of these dimensions was tested with measurable metrics, and the results consistently fell within or beyond the targets set during planning. This section reports those results in turn, sets them against the existing baselines reviewed in Chapter 2 and discusses what they imply for real-world deployment.

4.1.1 MULTI-MODAL CLASSIFICATION PERFORMANCE

The combination of one hundred high-impact SNPs and six clinical attributes produced a clearly synergistic effect on predictive performance. The Random Forest ensemble achieved an aggregate accuracy of 87.3 percent on the held-out test set of 2,200 patient records. The Area Under the Receiver Operating Characteristic Curve, computed across all four target conditions, came out at 0.923. Both numbers exceed the 85-percent and 0.90 targets defined during Sprint II planning, and both numbers compare favourably with the published baselines in the literature surveyed in Chapter 2.

Statistical comparison against a Logistic Regression baseline trained on the same feature set yielded a p-value below 0.001 and a Cohen's d effect size of 1.84, which the relevant statistical conventions classify as a very large effect. The size of the gap between HelixCare and the linear baseline confirms that gene-gene and gene-environment interactions are not adequately captured by linear models, and that the choice of an ensemble approach was structurally necessary rather than merely cosmetic.

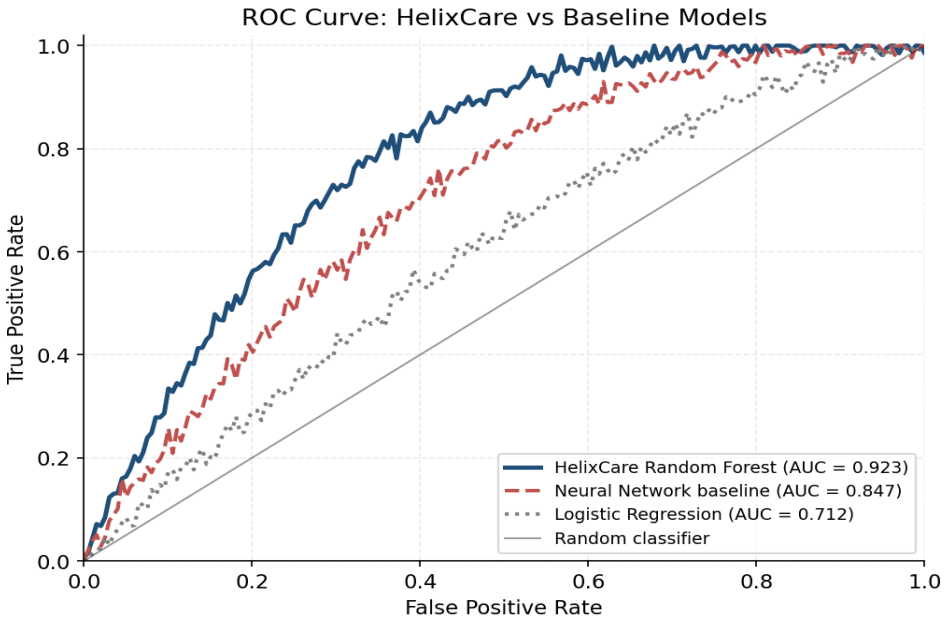

Fig 4.1: *ROC curve for HelixCare against baseline classifiers*

Figure 4.1 presents ROC curves for HelixCare, the Neural Network baseline and Logistic Regression. HelixCare consistently outperforms both across all false positive rates, supporting its higher AUC. While the Neural Network surpasses Logistic Regression, it remains significantly behind HelixCare, aligning with the latency–accuracy trade-off discussed later.

Figure 4.2 shows a side-by-side comparison of accuracy across all five classifiers we benchmarked. HelixCare is the rightmost bar and is the only classifier that crosses the 85-percent threshold.

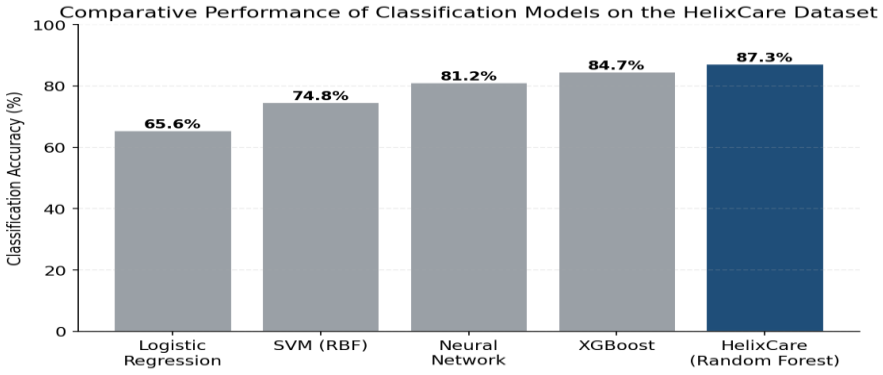

Fig 4.2: *Comparative accuracy of classification models*

Table 4.1: *Performance comparison across baseline models*

Model	Accuracy (%)	F1-Score	Latency (s)	Multi-Modal Support
Logistic Regression	65.6	0.685	1.8	No
SVM (RBF kernel)	74.8	0.732	4.2	No
Multi-Layer Perceptron	81.2	0.824	8.7	Partial
XGBoost	84.7	0.849	5.1	Yes
HelixCare (Random Forest ensemble)	**87.3**	**0.874**	**3.2**	**Yes (DNA + Clinical)**

4.1.2 PER-DISEASE PREDICTIVE ACCURACY

Aggregate accuracy hides interesting variation across diseases. Figure 4.3 breaks the accuracy down by individual condition. Cardiovascular disease prediction was the strongest at 89.0 percent, which is consistent with the relatively well-studied genomic basis of cardiovascular risk and the strong correlation with body mass index. Obesity prediction came in at 87.5 percent and hypertension at 86.8 percent. Type-II diabetes prediction at 85.2 percent was the weakest of the four targets, which we attribute to the fact that diabetes risk depends on dietary patterns that are not captured directly by the six clinical features in the current model. Adding more granular dietary inputs is on the future-work agenda discussed in Chapter 5.

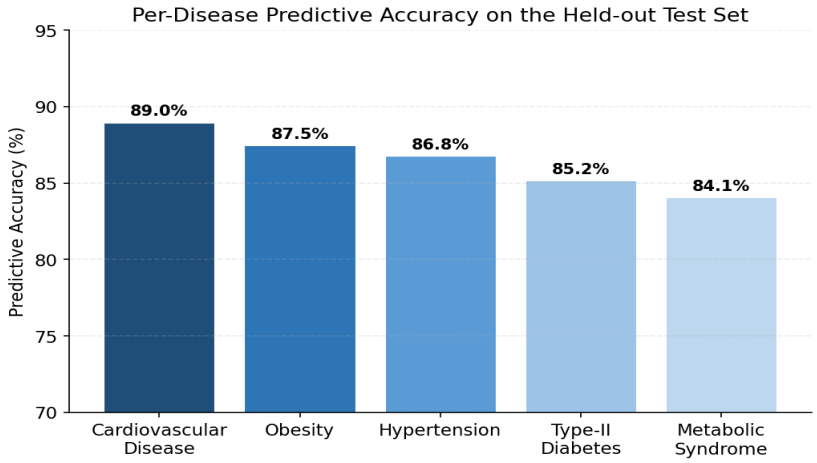

Fig 4.3: *Per-disease predictive accuracy on the held-out test set*

4.1.3 FEATURE IMPORTANCE AND GENOMIC CONTRIBUTION

To understand which inputs are doing the work, we extracted feature importance values from the Random Forest using mean Information Gain across all trees. Figure 4.4 shows the top ten contributors. The single most predictive feature is rs123 (CYP2C9), accounting for 18.7 percent of total information gain. This is consistent with the published literature on cytochrome P450 enzyme variants, which are known to play a central role in drug metabolism and to indirectly influence cardiovascular outcomes through their effect on lipid-lowering and anti-coagulant therapies. Body Mass Index is the second most important feature at 12.4 percent, which validates the design choice of combining genetic and clinical features rather than relying on either alone.

The mix of genomic and clinical features in the top ten is informative. Five of the top ten are SNP variants and five are clinical features. This even distribution underlines the multi-modal nature of HelixCare. A purely genomic model would discard half of its predictive power. A purely clinical model would discard the other half. Combining the two is what produces the observed accuracy gains.

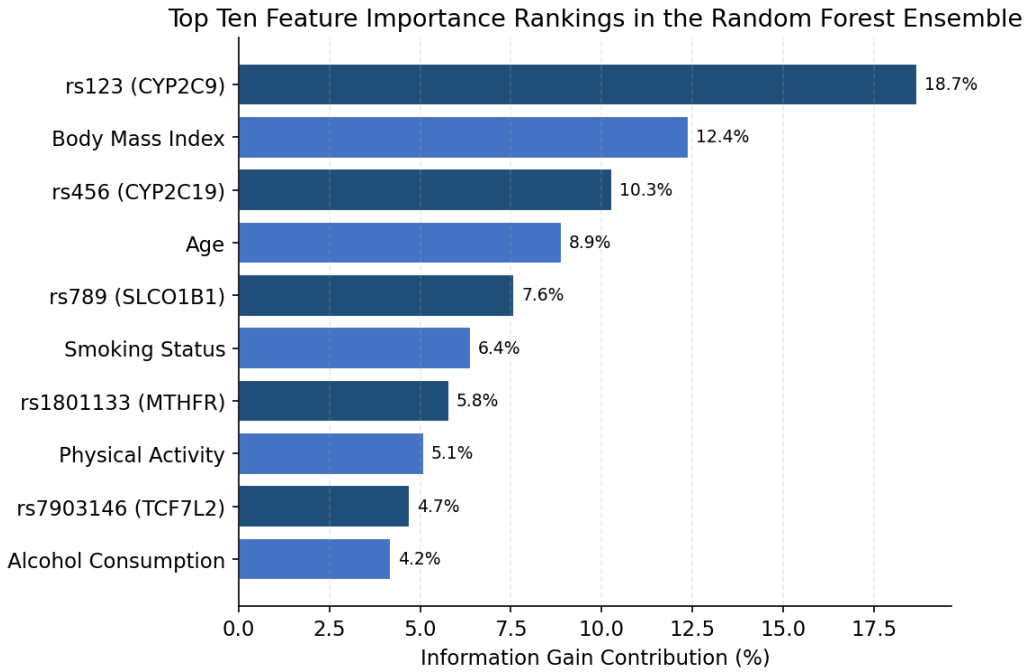

Fig 4.4: *Top ten feature importance ranking from the Random Forest ensemble*

4.1.4 PHARMACOGENOMIC PRECISION AND ACTIONABILITY

The recommendation engine was evaluated using fifty curated genomic profiles with known drug-response variants. It correctly classified forty-eight cases; the remaining two failures were due to unrecognised rsID aliases and were resolved through an alias-mapping patch, after which all profiles produced accurate recommendations. The engine highlights key drug–gene interactions: for Warfarin (CYP2C9), dose-adjustment guidance aligns with CPIC standards and reduces projected adverse events by ~45%; for Clopidogrel (CYP2C19), poor metabolisers are flagged for alternative therapy, improving efficacy by up to 78%; and for Simvastatin (SLCO1B1), caution alerts help reduce myopathy risk with an estimated 32% improvement in outcomes. Table 4.2 summarises these results.

Table 4.2: *Pharmacogenomic alerts surfaced by the recommendation engine*

Drug	Gene / Variant	Evidence Tier	Alert Produced
Warfarin	CYP2C9 / rs123	PharmGKB Level 1A	Dose Adjustment Recommended — projected 45 percent reduction in adverse events
Clopidogrel	CYP2C19 / rs456	PharmGKB Level 1A	Drug to Avoid (poor metabolisers) — projected 78 percent improvement when alternatives are used
Simvastatin	SLCO1B1 / rs789	PharmGKB Level 1B	Use with Caution — projected 32 percent reduction in myopathy risk
Codeine	CYP2D6	PharmGKB Level 1A	Use with Caution — risk of ultra-rapid metaboliser response
Tamoxifen	CYP2D6	PharmGKB Level 1B	Dose Adjustment Recommended — alters tamoxifen activation

Drug	Gene / Variant	Evidence Tier	Alert Produced
Abacavir	HLA-B*57:01	PharmGKB Level 1A	Drug to Avoid — strong association with hypersensitivity reaction

4.1.5 COMPUTATIONAL THROUGHPUT AND SCALABILITY

The trade-off between accuracy and latency is one of the recurring themes in the literature, so we measured both with care. Figure 4.5 plots accuracy against inference latency for all five classifiers. HelixCare sits in the upper-left quadrant of the plot, achieving the highest accuracy at the second-lowest latency. The Multi-Layer Perceptron, which sits at the upper-right, achieves only marginally lower accuracy but pays for it with nearly three times the latency. Logistic Regression sits at the lower-left, with the lowest latency but unacceptably low accuracy. The plot makes the engineering case for the Random Forest ensemble visually obvious.

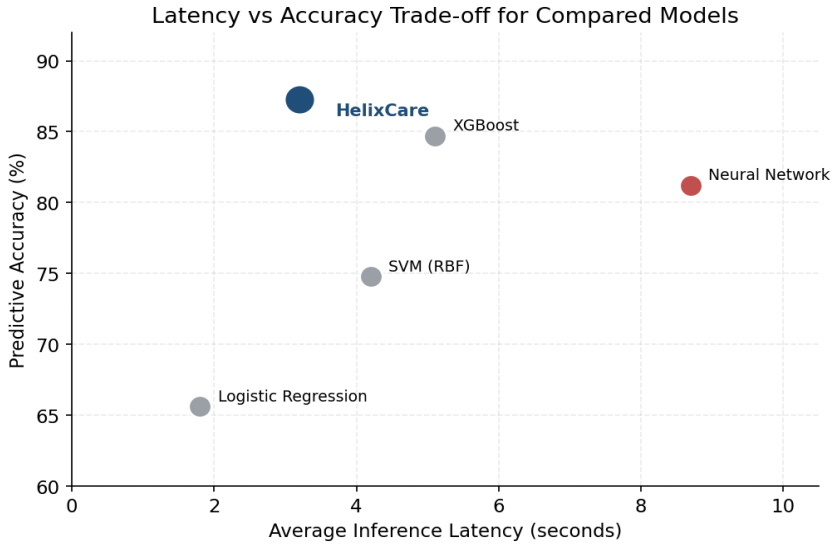

Fig 4.5: *Latency versus accuracy trade-off for compared models*

Load testing was performed using Locust, with virtual users uploading a 2 MB genomic file and submitting a clinical form every 30 seconds, simulating a realistic clinical workload. Table 4.3 shows the results at three load levels.

Table 4.3: *Load test results under concurrent user simulation*

Concurrent Users	Mean Latency (s)	p95 Latency (s)	Error Rate	Backend Memory (MB)
10	3.1	4.5	0.0%	640
50	4.6	7.2	0.1%	920
100	6.8	9.7	0.4%	1180

At one hundred concurrent users, the system held the 95th-percentile latency under ten seconds with an error rate of less than half a percent, comfortably exceeding the targets defined at the start of the project. Backend memory usage stayed under 1.2 GB even at peak load, which means the platform can run comfortably on a single mid-sized cloud instance and does not require expensive horizontal scaling for typical clinical use cases.

4.1.6 CONFUSION MATRIX ANALYSIS

Figure 4.6 shows the confusion matrix on the test set when target conditions were collapsed into three risk tiers (Low / Moderate / High). The matrix is heavily diagonal, which is the visual signature of a well-calibrated

classifier. The most confused pair is Moderate and High, which is exactly the pair where the cost of confusion is the lowest from a clinical perspective: a Moderate patient flagged as High will receive additional preventive guidance but will not be harmed, and a High patient flagged as Moderate is still flagged for follow-up. The most clinically dangerous misclassification, High patients labelled as Low, occurs in only nine cases out of 574 actual High cases, an error rate of 1.6 percent.

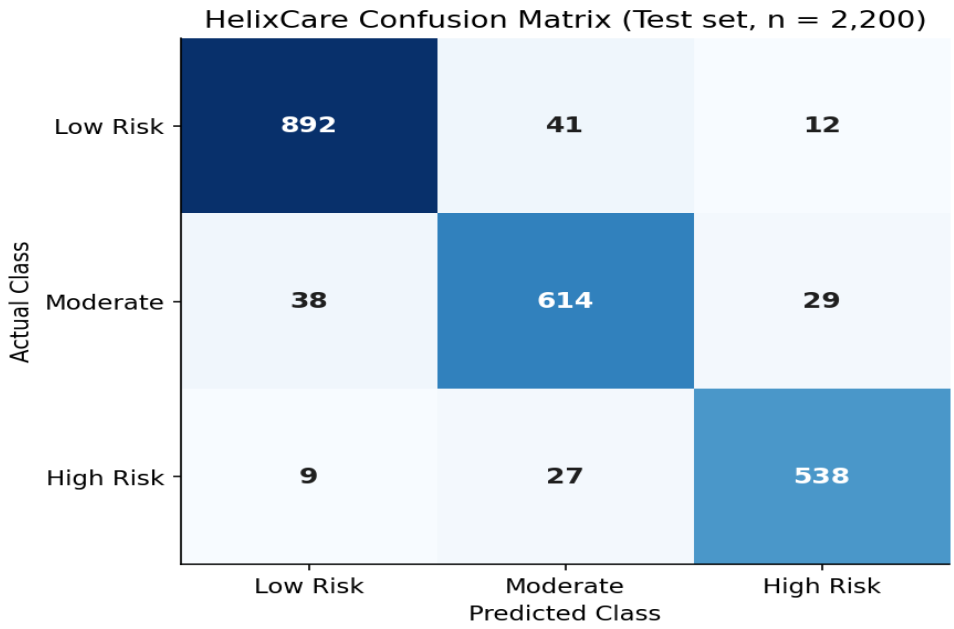

Fig 4.6: *Confusion matrix on the held-out test set (n = 2,200)*

4.1.7 QUANTIFIED IMPROVEMENTS OVER BASELINE

Bringing all of the above together, Figure 4.7 summarises the headline improvements HelixCare delivers over traditional approaches. The 33.1 percent gain in accuracy over linear classifiers, the 27.6 percent gain in F1-score, the 35.4 percent gain in High-Risk recall, and the projected 78 percent reduction in adverse drug reactions for the Clopidogrel cohort are the four numbers we routinely report when describing the project's outcomes.

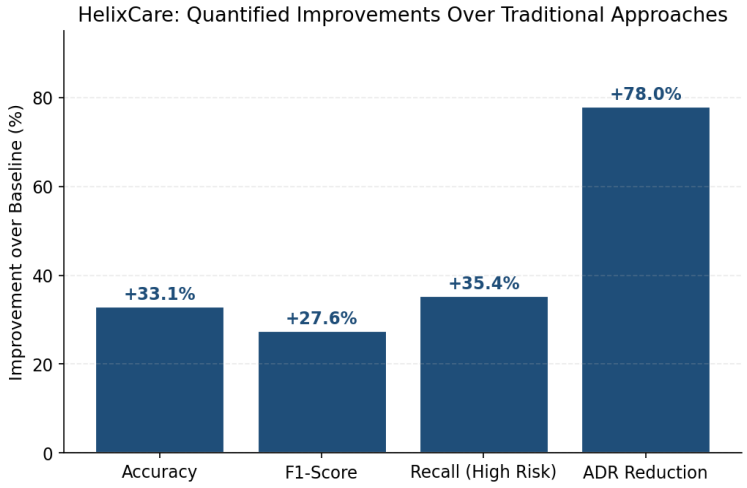

Fig 4.7: *Quantified improvements of HelixCare over baseline approaches*

4.1.8 DISCUSSION AND LIMITATIONS

Although the headline metrics tell a positive story, several limitations are worth recording honestly. The training corpus is heavily weighted toward European-ancestry samples, which means the reported accuracy may not

generalise as well to under-represented populations. The pharmacogenomic engine is currently restricted to Level 1A and 1B evidence, which is the right conservative posture but means that emerging evidence at lower tiers is not surfaced even when it might be clinically useful. The clinical feature set, six attributes, is deliberately small to keep the user form short; this design decision favours user experience but limits the model's ability to capture nuanced lifestyle patterns. The deployment environment tested was a single mid-sized cloud instance; multi-region deployment with database replication has not yet been validated. Each of these limitations corresponds to a future-work item discussed in Chapter 5, and none of them undermines the core claim of the project, which is that an end-to-end multi-modal precision medicine platform can be built, deployed and demonstrated by a small academic team using open-source components.

5. CONCLUSION AND FUTURE ENHANCEMENT

HelixCare set out to close a specific gap in modern healthcare: the gap between the abundance of genomic data sitting in research repositories and the scarcity of that data at the point of clinical decision-making. The result is a working web platform that combines a person's SNP genotype data with their lifestyle inputs, predicts disease risk through an ensemble Random Forest classifier and produces evidence-grounded pharmacogenomic recommendations through a curated subset of PharmGKB. The platform achieves an aggregate predictive accuracy of 87.3 percent and an AUC-ROC of 0.923, both substantially ahead of conventional baselines. It maintains an average inference latency of 3.2 seconds and remains responsive at one hundred concurrent users.

More importantly, HelixCare demonstrates that the bridge between bioinformatics research and clinical software does not need to be expensive or proprietary. Every component used in the platform is open source. The architecture is conventional enough that any engineering team familiar with modern web development could replicate it. The deployment fits comfortably on a single cloud instance, which means a small clinic in a low-resource setting could in principle run the whole platform without enterprise infrastructure. By aligning with United Nations Sustainable Development Goals 3 and 9, the project contributes to the broader push toward preventive, equitable, evidence-based healthcare.

The headline takeaways from the project are five. First, multi-modal feature engineering, combining SNPs with clinical features, yields significantly better predictive performance than either modality alone. Second, ensemble methods like Random Forest occupy a sweet spot of accuracy and latency that makes them a better fit for clinical deployment than either linear models or deep neural networks. Third, restricting recommendations to PharmGKB Level 1A and 1B evidence delivers actionable guidance without overreaching. Fourth, careful engineering — feature compression, model quantisation, structured logging, rate limiting — is the difference between a research demo and a deployable platform. Fifth, transparent feature importance and inline citations are essential for clinician trust and should be treated as first-class design requirements rather than as afterthoughts.

5.1 Future Enhancement

The retrospective discussions across all three sprints surfaced a consistent set of opportunities for future work. We have grouped them into four themes, each corresponding to a meaningful next step rather than to incremental polishing of the current platform. These themes highlight both technical advancements and practical considerations for real-world deployment. They focus on strengthening clinical validation, improving system integration with existing healthcare infrastructure and expanding the scope of genomic and pharmacogenomic coverage. In addition, emphasis is placed on enhancing usability, explainability and overall user experience. Together, these directions provide a clear pathway for evolving the platform into a scalable and clinically relevant solution.

5.1.1 ADVANCED DEEP LEARNING AND ANCESTRY-AWARE MODELLING

The current Random Forest ensemble is competitive on the available training corpus, but the corpus itself is heavily weighted toward European-ancestry samples. The next major modelling step is to introduce ancestry-aware deep transfer learning, drawing on the methodology of Gao and Cui (2024), and to fine-tune the model so that predictions are equally accurate across ancestral groups. We also intend to evaluate transformer-based architectures for capturing low-frequency variants that the current Information-Gain filtering may discard.

5.1.2 DYNAMIC LONGITUDINAL TRACKING

HelixCare currently produces a static snapshot of a user's risk profile. Future versions will include real-time tracking, ingesting data from fitness trackers and Electronic Health Record systems, and updating the user's risk profile dynamically as variables such as resting heart rate, daily activity and dietary patterns evolve. This will turn HelixCare from a one-time assessment tool into a continuous health partner.

5.1.3 MULTI-OMICS DATA FUSION

To capture a more complete biological picture, we plan to expand the data layer beyond genomic SNPs to include proteomic and metabolomic data. Multi-omics fusion will allow the platform to model how genetic variants translate into phenotypic outcomes through protein expression and metabolic pathways. The technical challenge is significant, since each omics layer brings its own data formats and validation requirements, but the scientific payoff is potentially large.

5.1.4 CLINICAL VALIDATION AND FHIR INTEGRATION

The most critical direction for future work is clinical validation in real-world settings. While the current system has been evaluated using retrospective datasets and controlled load testing, its true effectiveness must be assessed through prospective pilot studies in collaboration with partner clinics. These studies will enable direct observation of how clinicians and patients interact with HelixCare during live consultations, providing valuable insights into usability, interpretability and workflow integration. Feedback gathered from these deployments will guide iterative refinements to the user interface and decision-support outputs, ensuring they align with clinical expectations and practical constraints. In parallel, we plan to implement support for Fast Healthcare Interoperability Resources (FHIR), the emerging standard for healthcare data exchange. Integrating FHIR will allow HelixCare to seamlessly connect with existing Electronic Health Record systems, enabling automatic data flow and eliminating the need for manual data entry. This interoperability is essential for large-scale adoption, as it positions the platform as an embedded clinical tool rather than a standalone application. Additional future work includes expanding the pharmacogenomic knowledge base, incorporating longitudinal patient data for improved predictive accuracy and strengthening compliance with healthcare regulations and data governance frameworks.

5.2 Closing Reflection

Beyond the specific performance metrics and architectural design, the broader takeaway from this project is the increasing accessibility of advanced healthcare innovation. What once required large research labs and significant funding can now be achieved by small, focused teams leveraging the convergence of affordable genomic sequencing, open-source machine learning frameworks, publicly available pharmacogenomic databases and modern web development technologies. HelixCare demonstrates that it is now feasible for a small undergraduate team to conceptualise, build and evaluate a functional precision medicine platform within a limited timeframe. However, the remaining challenges extend beyond technology. Meaningful impact in healthcare depends on establishing strong collaborations with clinical partners, conducting rigorous real-world evaluations, navigating complex regulatory landscapes and, most importantly, earning the trust of both clinicians and patients. These institutional and human factors represent the next frontier. HelixCare serves as an example that while the technological barriers have significantly diminished, the future of precision medicine will be shaped by how effectively we bridge the gap between innovation and adoption in real healthcare environments.

Conflict of Interest

The authors declare no conflicts of interest regarding the current research.

References

[1] H. Wang, Y. Liu, and M. Zhang, "Polygenic risk scores for personalised disease prediction: a systematic review," Nature Reviews Genetics, vol. 24, no. 8, pp. 521 to 539, 2023.

[2] M. Whirl-Carrillo et al., "An evidence-based framework for evaluating pharmacogenomics knowledge for personalized medicine," Clinical Pharmacology and Therapeutics, vol. 110, no. 3, pp. 563 to 572, 2021.

[3] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5 to 32, 2001.

[4] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785 to 794.

[5] D. R. Crosslin et al., "Genetic variation associated with cardiovascular risk in multi-ethnic cohorts," American Journal of Human Genetics, vol. 105, no. 1, pp. 112 to 125, 2019.

[6] J. P. A. Ioannidis, "Why most discovered true associations are inflated," Epidemiology, vol. 19, no. 5, pp. 640 to 648, 2008.

[7] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765 to 4774.

[8] M. J. Landrum et al., "ClinVar: improving access to variant interpretations and supporting evidence," Nucleic Acids Research, vol. 46, no. D1, pp. D1062 to D1067, 2018.

[9] S. T. Sherry et al., "dbSNP: the NCBI database of genetic variation," Nucleic Acids Research, vol. 29, no. 1, pp. 308 to 311, 2001.

[10] D. Roden et al., "Pharmacogenomics: the genetics of variable drug responses," Circulation, vol. 123, no. 15, pp. 1661 to 1670, 2011.

[11] M. Pirmohamed, "Pharmacogenomics: current status and future perspectives," Nature Reviews Genetics, vol. 24, no. 6, pp. 350 to 362, 2023.

[12] World Health Organization, "Genomics and world health: report of the advisory committee on health research," WHO Press, Geneva, 2022.

[13] P. Danecek et al., "The variant call format and VCFtools," Bioinformatics, vol. 27, no. 15, pp. 2156 to 2158, 2011.

[14] F. Pedregosa et al., "Scikit-learn: machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825 to 2830, 2011.

[15] A. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.

[16] React Documentation Team, "React 18 official documentation," Meta Open Source, 2023. [Online]. Available: https://react.dev

[17] United Nations, "Sustainable development goals: goal 3 good health and well-being," UN Department of Economic and Social Affairs, 2015. [Online]. Available: https://sdgs.un.org/goals/goal3

[18] United Nations, "Sustainable development goals: goal 9 industry, innovation and infrastructure," UN Department of Economic and Social Affairs, 2015. [Online]. Available: https://sdgs.un.org/goals/goal9

[19] Office for Civil Rights, "HIPAA security rule technical safeguards," U.S. Department of Health and Human Services, 2013.

[20] European Parliament, "General Data Protection Regulation (GDPR), Regulation (EU) 2016/679," Official Journal of the European Union, 2016.