

I. Project Framework: Context, Initiative, and Goal

The Clinical Context

In 2026, Chronic Kidney Disease (CKD) remains a critical public health challenge where late-stage detection is the primary obstacle to treatment. Because early kidney damage is often asymptomatic, patients frequently progress to irreversible stages before diagnosis. In busy clinical settings, urologists and primary care doctors are overwhelmed with scattered laboratory data. They are forced to manually synthesize creatinine levels, urine protein ratios, and historical trends under significant time pressure, leading to "clinical inertia"—where dangerous downward trajectories in kidney function are overlooked because they are not visualized.

The Product Initiative:

To address this gap, your team is developing **CKD-EarlyInsight**, a specialized Clinical Decision Support System (CDSS) for the modern healthcare environment. This initiative moves beyond simple risk calculators by providing a "proactive audit" of patient health. The platform identifies "Care Gaps," such as a diabetic patient missing their annual urine protein screen, and automatically calculates the eGFR "slope"—the rate at which a patient's kidney function is declining over time. This provides clinicians with a visual narrative of the disease rather than a single, isolated data point.

The Project Goal

The goal of this capstone is to deliver a fully functional, cloud-hosted, and **SMART on FHIR**-compatible web application. By the end of the semester, the product will be capable of ingesting population data from the NHANES 2017-2018 cycle and re-classifying patients using the **2021 race-free CKD-EPI** medical standard. The app will provide **Explainable AI (SHAP)** insights, telling the doctor exactly which factors—such as age, blood pressure, or lab results—are driving an individual patient's risk profile.

II. Technical Team Roles

- **Member 1: ML & Clinical Logic Lead**

This member owns the "brain" of the application. They are responsible for translating the **2021 CKD-EPI medical formulas** into Python code and developing the machine learning models that predict progression. Their focus is on clinical accuracy and integrating **SHAP** (Shapley Additive Explanations) to provide transparency, so the app can explain *why* it flagged a specific patient as high-risk.

- **Member 2: Backend & Data Engineer**

This member is the architect of the data pipeline and server infrastructure. They manage the heavy lifting of downloading, cleaning, and merging the five NHANES data files. They are responsible for building the **PostgreSQL** database and developing the **FastAPI** backend that allows the frontend to communicate with the clinical logic, ensuring that patient data is retrieved in milliseconds.

- **Member 3: Frontend & UX Engineer**

This member is responsible for the doctor's interface. They design and build the interactive dashboard, including the **KDIGO Heatmap** (the color-coded clinical standard) and longitudinal trend charts using libraries like **Plotly** or **D3.js**. Their task is to ensure complex medical data is presented intuitively, allowing a clinician to understand a patient's kidney history at a glance.

- **Member 4: DevOps & Interoperability Engineer**

This member ensures the product is stable, secure, and integrated. They manage **Docker** containerization and cloud deployment (AWS/Azure). Their primary focus is the **SMART on FHIR** implementation, ensuring the app can ingest data from a simulated Electronic Health Record (EHR) using the FHIR API standard.

- **Member 5: Validation & Quality Engineer**

This member is responsible for **Automated Testing** and **Clinical Validation**. They do not "document" the project; they write the code that *tests* the project. They develop unit tests for the eGFR formulas, load tests for the API, and validation scripts that compare the app's output against the National Kidney Foundation calculators to ensure 100% mathematical accuracy.

III. 7-Sprint Technical Execution Roadmap

Sprint 1: The Infrastructure Foundation (Weeks 1-2)

Member 2 downloads the NHANES DEMO_J, BIOPRO_J, ALB_CR_J, BPX_J, and DIQ_J files and merges them via SEQN. Member 1 prototypes the 2021 race-free eGFR formula. Member 4 initializes the GitHub repository, sets up the Docker environment, and Member 3 builds the high-fidelity UI mocks. Member 5 builds the **Test Suite Environment**, setting up the framework for automated unit testing in GitHub Actions.

Sprint 2: Clinical Engine & DB Integration (Weeks 3-4)

Member 1 applies the eGFR formula to the entire merged NHANES dataset. Member 2 hosts this data in a PostgreSQL database and creates the initial API endpoints. Member 3 builds the static KDIGO Risk Heatmap UI. Member 4 deploys the first "live" version of the container to the cloud. Member 5 writes the first **Validation Scripts**, ensuring the database merge didn't lose any critical patient records.

Sprint 3: API Logic & Trend Visualization (Weeks 5-6)

Member 2 expands the API to serve patient-specific histories. Member 1 codes the "Care Gap" logic to flag missing uACR tests. Member 3 implements the **eGFR Trend Charts**, enabling the visualization of kidney function "slopes." Member 4 implements **Auth0** for secure login. Member 5 writes the **Logic Tests**, verifying that the "Care Gap" flags trigger correctly for diabetic patients in the dataset.

Sprint 4: AI Integration & FHIR Interoperability (Weeks 7-8)

Member 1 trains the risk model and integrates the **SHAP** library for model explainability. Member 2 updates the API to send SHAP values to the UI. Member 3 builds the "Risk Driver" panel in the dashboard. Member 4 connects the app to the **SMART on FHIR Sandbox** to prove EHR compatibility. Member 5 performs **Cross-Validation**, comparing the ML model's risk scores against standard clinical risk calculators.

Sprint 5: Performance & Security Hardening (Weeks 9-10)

Member 2 optimizes database indices to ensure sub-500ms search times. Member 3 ensures the dashboard is fully responsive for tablets. Member 4 configures SSL/HTTPS and performs a security audit on the cloud environment. Member 1 finalizes the SHAP global risk visualizations. Member 5 executes **Load Testing**, simulating multiple simultaneous clinician sessions to ensure the backend doesn't crash under pressure.

Sprint 6: Final Polish & Production Audit (Weeks 11-12)

Member 4 records the final product demonstration, specifically showing the app launching from a FHIR sandbox. Member 3 adds final branding and polish to the UI. Member 2 and Member 1 conduct a final clinical audit of the data. Member 5 performs a **Final Regression Test**, ensuring that recent UI changes didn't break the original clinical formulas or the database connections.

Sprint 7: Release & Presentation (Final Week)

Member 4 creates the final **v1.0 Release Tag** on GitHub. Members 2 and 3 stabilize the live production environment for the demonstration. Member 1 and Member 5 finalize the technical charts and model performance metrics. The entire team collaborates on the final presentation, focusing on a live demo that shows a patient being moved from "undiagnosed" to "flagged" by the tool.

IV. GitHub Repository Structure

text

```
ckd-early-insight/
├── .github/workflows/    # CI/CD & Automated Testing (M4 & M5)
├── app/                  # Frontend code (M3)
└── api/                  # Backend API & SQL Schemas (M2)
```

— logic/	# eGFR Formula & SHAP Models (M1)
— fhir/	# SMART on FHIR integration scripts (M4)
— tests/	# Automated Unit, Logic, & Load Tests (M5)
— Dockerfile	# Production Container Config (M4)
— README.md	# Technical setup & installation guide

CKD-EarlyInsight platform using the gold-standard 2017-2018 NHANES cycle, your team will need to ingest five specific datasets. These must be downloaded as SAS Transport (.XPT) files and merged using Python.

I. Primary Dataset Links ([NHANES 2017-2018](#))

Member 2 (Backend & Data) should download these directly into the /data folder of your repository.

- **Demographics (DEMO_J):** Essential for Age and Gender (used in eGFR calculations).
- **Standard Biochemistry (BIOPRO_J):** Contains **Serum Creatinine**, the core variable for kidney function.
- **Urine Albumin & Creatinine (ALB_CR_J):** Contains **uACR** data, required to identify early-stage damage and "Care Gaps."
- **Blood Pressure (BPX_J):** Provides hypertension data for the risk prediction model.
- **Diabetes Questionnaire (DIQ_J):** Used to identify diabetic patients who are missing required annual kidney screenings.

II. Clinical & Interoperability Tools

These tools are required for Member 1 (ML/Logic) and Member 4 (DevOps) to ensure the app meets 2026 medical and technical standards.

- **2021 CKD-EPI Race-Free Equation:** This is the mandatory clinical formula Member 1 must implement in Python to calculate kidney function (eGFR).

- [SMART App Launcher \(FHIR Sandbox\)](#): Member 4 will use this to test the app's ability to launch from a simulated Electronic Health Record (EHR) like Epic or Cerner.
 - **KDIGO CKD Heat Map Reference**: Member 3 (Frontend) should use the "Prognosis of CKD by GFR and Albuminuria Categories" chart (Figure 1 in the link) to design the dashboard's risk grid.
 - **NKF eGFR Calculator (Validation Tool)**: **Member 5** (Validation) will use this manual calculator to verify that the app's automated math exactly matches official clinical outputs.
-

III. Implementation Code for Member 1 & 2

To get the project moving in Sprint 1, Member 2 can use this Python snippet to merge the five files into a single master database:

```
import pandas as pd

# Load files (Ensure you have 'pyreadstat' or 'xport' installed)
demo = pd.read_sas('DEMO_J.XPT')
biochem = pd.read_sas('BIOPRO_J.XPT')
urine = pd.read_sas('ALB_CR_J.XPT')
bp = pd.read_sas('BPX_J.XPT')
diabetes = pd.read_sas('DIQ_J.XPT')

# Merge all on the unique Sequence Number (SEQN)
df = pd.merge(demo, biochem, on='SEQN')
df = pd.merge(df, urine, on='SEQN')
df = pd.merge(df, bp, on='SEQN')
df = pd.merge(df, diabetes, on='SEQN')

# Member 1 now has thousands of records to apply the 2021 formula to
print(f'Dataset successfully merged. Total records: {len(df)}')
```

IV. Summary of Technical Roles for Success

- **Member 1**: Use the **NKF 2021 Formula** link to build the logic.

- **Member 2:** Use the **NHANES 2017-2018** links to build the database.
- **Member 3:** Use the **KDIGO Heat Map** link to build the UI visuals.
- **Member 4:** Use the **SMART App Launcher** link to ensure the app is "EHR-ready."
- **Member 5:** Use the **NKF eGFR Calculator** (on the same page as the formula) to write tests that verify the app's math is correct. National Kidney Foundation Calculator