CSC 230 High Level Design

- What are the major logical components of your system?
    - User Interface: React and Tailwind
    - API Gateway: Node.js and Express
    - AI Engine: OpenAI (most likely GPT 4.0)
        - Logic layer that will suggest models and generate glossary/tooltips
    - R Sandbox: Plumber
        - Executes regression templates
    - Glossary Store: Markdown/JSON
        - Knowledge base that should be accessible from the UJI and API
    - Storage
        - Should save project files and results, as well as the uploaded cvs/xlsx files
    - Access Control:
        - Has not been discussed as on yet
- What is the role/purpose of each component of your system?
    - UI
        - Single-page app to upload CSV/XLSX, show detected variable types, present suggested model, display R code with results and marginal-effect/diagnostic visuals, also inline glossary tooltips.
    - API (via Express)
        - Single entry point. AObjective is to validate uploads, manages sessions, orchestrates calls to AI and R, and should return JSON for UI.
    - AI Engine
        - Detects/labels variable types.
        - Suggests the appropriate regression family (OLS/Logistic/Poisson/NegBin/Ordered)
        - Generates/explains glossary strings.
        - Guardrails: template-only execution! no free-form code (bad robot)
    - R Sandbox (Plumber)
        - Runs curated R templates; returns broom-tidied JSON (coefficients, SEs, p-values, fit stats; optional robust SEs/VIF).
    - Glossary Store
        - Markdown/JSON terms (p-value, VIF, logistic vs. OLS, etc.) surfaced in UI as tooltips.
    - Storage
        - Saves uploads (temporary), runs configs, displays JSON results; enables re-runs and demos.
    - Access Control: I dunno
- What tools / technologies / programming languages are you using to build each component?
    - UI
        - React w/ Vite and Tailwind

    - API
        - Node.js w/ Express; Multer (uploads)
        - CORS
        - Jest (tests)
        - Docker/Compose
    - AI Engine
        - OpenAI tools
            - for suggestions/glossary
            - rules/templates owned by Integration/Testing dev.
    - R Sandbox:
        - R /plumber/broom/car/sandwich/MASS
    - Docs/KB
        - Markdown and Git (Knowledge Base Curator).
    - Local run
        - Docker/Compose (for all services)
- Which components of the system communicate/interact with each other? (E.g., If you have a database, which components of the system talk to the data)
    - From UI to API (via Express):
        - POST /api/upload (for cvs/xlsx files)
        - POST /api/suggest (from an outlined summary, to suggesting models)
        - POST /api/run (runs models and options for model)
        - GET /api/result/:id (JSON and glossary)
    - 2-way comms between API and AI Engine
        - Send outline and features, returns variable types, model suggestions, explanations and glossary text
    - 2 way comms between API and R sandbox (via Plumber)
        - Send template parameters, receives JSON
    - 2 way comms between  API and Storage component
        - Uploads, outputs
    - 2 way comms between UI and Glossary Store (using API)
        - Term lookup for tooltips