Acme AI Fellowship (Cohort V)

Admin-Driven Rule-Engine Utility Bill Calculator

Build a simple full-stack web application that calculates a utility bill (e.g., electricity) based on a configurable flat-rate rule. An admin can update the rate and global constants, and end users submit consumption data and receive an accurate bill calculation using the active rule, without any visibility into the rules or logic

This assignment is intentionally scoped to be completed within 8-10 hours and focuses on core full-stack fundamentals, deployment, and clean UI not advanced business logic.

Tech Stack
- **Frontend**: React (TypeScript recommended)
- **Backend**: NestJS (TypeScript)
- **Database**: PostgreSQL (free hosted service, e.g., Supabase, Neon, Render)
- **Deployment**: Mandatory - Frontend (Vercel/Netlify), Backend (Render/Railway)

Project Overview
The system has two simple interfaces:
1. Admin Interface - update pricing configuration
2. User Interface - calculate utility bills

Only one flat-rate rule exists at any time, and users do not see any rule logic- only the final calculation result.

Tasks

Module	Requirement Description
Admin Configuration	The system shall allow an admin to update the rate per unit, VAT percentage, and fixed service charge through an admin interface.
Admin Access Control	The admin interface shall be protected using a simple access mechanism (e.g., admin key or PIN).
Configuration Storage	The system shall store the latest pricing configuration in the database, maintaining only one active configuration record.
User Input	The system shall allow users to input the number of units consumed for bill calculation.
Bill Calculation	The system shall calculate the subtotal, VAT amount, service charge, and total payable amount based on the latest configuration.
Result Display	The system shall display the calculation results clearly to the user in a structured format.
Configuration Retrieval	The system shall retrieve the latest configuration dynamically when performing calculations.
Validation	The system shall validate inputs (e.g., units must be a positive number; VAT between 0–100).
API Endpoints	The system shall expose REST APIs to fetch configuration, update configuration, and perform bill calculations.
Deployment	The system shall be deployed with a publicly accessible frontend, backend, and database.
PDF Generation	Display the result clearly and enable download as PDF

Evaluation Criteria

Criteria	Weight	Description
Functionality	High	All required features work correctly (rule activation, calculation, download)
Code Quality	High	Clean, modular, typed, well-commented code with good practices
Architecture & Design	High	Proper separation (modules, services, DTOs), good API design, and DB schema
UI/UX	Medium	Simple, clear, and usable interface
Deployment	High	Live URLs work correctly
Documentation (README)	Medium	Clear setup, architecture overview, deployment links, assumptions

Submission Guidelines

- Deadline: 24 November 2025, by 11:00 AM
- Deliverable: A single PDF document containing all answers to the test questions
- File Name: Fellowship-C5-AAI-[Your Full Name].pdf
- Submission Method: Upload the PDF via the Google Form linked here

Submission Format (PDF only)

- All tasks, code (as text), live deployment links, screenshots, architecture overview and written explanations must be compiled into one single PDF document.
- Do not submit a GitHub link as the primary submission (you may include it as an optional reference link inside the PDF).

AI Tool Usage Policy

If any AI tools were used (ChatGPT/Copilot/Gemini/etc.), clearly mention what was generated and where it was used. Undisclosed AI-generated content may result in negative marking and/or disqualification.