Assignment Instructions

You are tasked with building the first version of an integration layer between a new core system (named **ThreadPilot**) and multiple legacy systems.

Scenario:

- Implement **two separate application projects** (for example, two microservices or APIs).
- **Endpoint 1:** Accepts a vehicle registration number and returns information about the vehicle.
- **Endpoint 2:** Accepts a personal identification number and returns all the insurances the person has, along with their monthly costs.
  - If the person has a car insurance, the car information should be included in the response.
  - *This means Endpoint 2 should integrate with Endpoint 1.*

Insurance products and prices:

- Pet insurance: $10
- Personal health insurance: $20
- Car insurance: $30

Requirements:

- Implement **at least two separate application projects** (not just two endpoints in a single project).
- Both endpoints must be accessible via REST APIs.
- At least **3 unit tests** (for key logic; more is better).
- **Graceful error handling** for invalid input or missing data.
- Include a **README.md** (or equivalent) that:
  - Explains your architecture and design decisions
  - Describes how to run and test the solution locally
  - Discusses your approach to error handling, extensibility, and (if applicable) security
- Code should be clean, maintainable, and follow good practices.

Bonus Points:

- **Wide test coverage** (beyond 3 tests): including integration tests or test strategy explanation.

- **Basic DevOps pipeline:** (e.g., sample YAML for CI, GitHub Actions, Azure DevOps, or equivalent—showing build and test steps).
- **Code architecture:** Use of patterns that make the solution extensible and maintainable.
- **Mocking/abstraction** for legacy systems or external dependencies.
- **Handling of edge cases** (e.g., missing vehicles, no insurances, multiple insurances).
- **Discussion of API versioning and future extensibility** in your documentation.
- (Optional but appreciated) Brief section on how you would approach onboarding or enabling other developers to work with your solution.

Personal Reflection (Required):

- At the end of your README, please include a **short reflection (3–5 sentences)** on:
  - Any similar project or experience you've had in the past.
  - What was challenging or interesting in this assignment.
  - What you would improve or extend if you had more time.

Submission:

- Package your code and documentation as a zip file or public Git repository.
- Submit your solution **no later than 24 hours before your interview slot**.