Test Assignment

Objective:

Develop an API endpoint that, based on a parameter passed in the URL, sends a request to one of two (potentially more in the future) external systems and returns a unified response.

Additionally, implement the same functionality for CLI. Based on the parameter passed in the CLI command, the server should send a request to one of two external systems and return a unified response in the console.

The request is a regular, one-time purchase, no 3DS, server-2-server.

External systems are Shift4 and ACI.

Technical Requirements:

- PHP 8
- Symfony 6.4

Bonus Points For:

- Implementing any tests (unit/integration/functional)
- Including a simple Dockerfile

Detailed Description:

Both CLI Command and API Endpoint:

1. Should accept input params:
    - amount
    - currency
    - card number
    - card exp year
    - card exp month
    - card cvv
2. Return a unified response regardless of which external system is called. The response should contain:
    - transaction ID
    - date of creating
    - amount
    - currency
    - card bin

API Endpoint exmaple: `/app/example/{aci|shift4}`

CLI Command example: `bin/console app:example {aci|shift4}`

Based on the parameter value `{aci|shift4}`, the server should send a request to one of two external systems.

External Systems:

Shift4

- Doc: https://dev.shift4.com/docs/api#charge-create
- These parameters could be hardcoded because they are limited to the test mode:
  - auth key
  - card number

ACI

- Doc: https://docs.oppwa.com/integrations/server-to-server#syncPayment
  - Synchronous payment → Debit → Perform debit payment
- These parameters could be hardcoded because they are limited to the test mode:
  - auth key
  - entity id
  - payment brand
  - card number
  - currency (EUR)

Expectations:

- The API endpoint and CLI command should be properly documented.
- The code should follow best practices and be well-organized.
- Ensure error handling and edge cases are considered.
- Use domain-specific terminology as provided in the attached documents.

Submission:

- Provide the complete source code in the public repository (GitHub / GitLab / Bitbucket)
- Include instructions on how to run the application and tests.
- Dockerfile (if implemented) should allow easy setup of the environment.