

Also, build its supporting backend API. This exercise will test your ability to design and implement both a frontend view and backend functionality under time constraints.

Requirements

1. Frontend: Items Page

Design & Functionality:

- Create an item detail page that mimics the look and feel of MercadoLibre.

The page should display key information such as:

- Product images
- Title and description
- Price
- Payment methods
- Seller information
- Additional details (e.g., ratings, reviews, or available stock)
- Recommendations isn't required
- Checkout logic is not required

User Experience:

- Ensure the page is responsive and user-friendly.

Stack:

- You can use any frontend technology or framework of your choice.

2. Backend: API Development

API Endpoints:

- Implement a RESTful API that supports the frontend by providing the necessary product details
- The primary endpoint should fetch product details

Stack:

- You can use any backend technology or framework of your choice
- Do not use real databases, persist everything in local JSON or CSV files

Non-functional requirements:

Proper error handling, documented, at least 80% code coverage

Tool Usage

Allowed Tools: You may use and are encouraged to use GenAI tools, agentic IDEs, and other code assistance tools to help generate ideas or code.

Note: The final submitted code should reflect your understanding and modifications. Be prepared to explain the solution.

3. Documentation:

Along with your code submission, include a brief (1-2 page) document that:

- Explains your design choices.
- Describes any challenges you faced and how you addressed them.

4. Submission

Provide a link to your repository or a zipped project folder.

It must contain a run.md explaining how to run the project

Live Coding Session (1 Hour)

Objective

Extend your solution by adding a new feature, applying discounts to product prices in the backend. This session is designed to assess your ability to adapt and extend your own work under observation.

Task

Feature Implementation: Modify your existing API endpoint (or create a new one) to include a discount mechanism. For example, the API response should now include a discounted price calculated from the base price.

Ensure the discount logic is flexible (allowing for different discount percentages or conditions).

You will be asked to share your screen and walk through your code and test creation process.

Discussion Points:

Explain your design and coding decisions

Discuss how the discount feature could be further enhanced or scaled

Evaluation criteria

Technical proficiency: Clear, maintainable code with proper error handling.

Correct implementation of the discount logic and its integration with the existing API.

Communication & explanation:

Clarity in explaining design choices and thought process

Insight into potential improvements and scalability considerations