

1. Viết một User Story cho một ứng dụng bán hàng trực tuyến, trong đó người dùng có thể thêm sản phẩm vào giỏ hàng.

Yêu cầu:

- Cấu trúc User Story phải có đầy đủ ba phần:
  - Role (Vai trò) – Ai là người cần chức năng này? (Người dùng, khách hàng, quản trị viên, v.v.)
  - Goal (Mục tiêu) – Họ muốn làm gì?
  - Benefit (Lợi ích) – Tại sao họ cần chức năng đó?

Ví dụ:

- "As a customer, I want to add products to my shopping cart so that I can purchase multiple items in one order."
  - Role: Customer (Khách hàng)
  - Goal: Add products to shopping cart (Thêm sản phẩm vào giỏ hàng)
  - Benefit: Purchase multiple items in one order (Mua nhiều món trong cùng một đơn)

As a customer, I want to add products to my cart, so I can continue to select more products and pay for them all at once.

- **Role:** Customer(Khách hàng)
- **Goal:** Add products to my cart(Thêm sản phẩm vào giỏ hàng của mình)
- **Benefit:** continue to select more products and pay for them all at once(Tiếp tục chọn thêm sản phẩm khác và thanh toán tất cả cùng lúc)

2. Viết một User Story cho một ứng dụng bán hàng trực tuyến, trong đó người dùng có thể thêm sản phẩm vào giỏ hàng.

Yêu cầu:

- Viết một User Story cho một ứng dụng thương mại điện tử, trong đó người dùng có thể thanh toán đơn hàng của mình qua các phương thức

như thẻ tín dụng hoặc ví điện tử.

Yêu cầu:

Cấu trúc User Story phải có đầy đủ ba phần:

- Role (Vai trò) – Ai là người cần chức năng này?
- Goal (Mục tiêu) – Họ muốn làm gì?
- Benefit (Lợi ích) – Tại sao họ cần chức năng đó?

As a customer, I want to pay for my order using multiple methods, so that I can be flexible in paying and completing my purchase.

- **Role:** Customer(Khách hàng)
- **Goal:** Pay for my order using multiple methods(thanh toán đơn hàng của mình bằng nhiều phương thức)
- **Benefit:** Be flexible in paying and completing my purchase(Linh hoạt trong việc thanh toán và hoàn tất đơn hàng)

### 3. Viết một Product Backlog cho một ứng dụng quản lý công việc, nơi người dùng có thể tạo, quản lý, và theo dõi các công việc của mình.

Yêu cầu:

Liệt kê ít nhất 5 Product Backlog Items (PBI) cho các tính năng cần có trong ứng dụng.

Mỗi PBI phải có tên, mô tả ngắn gọn về chức năng, độ ưu tiên và ước lượng điểm Story Points (nếu có thể).

Product Backlog phải đảm bảo sự rõ ràng về các tính năng quan trọng cần được triển khai trong các Sprint tiếp theo.

Ví dụ:

- Tạo công việc mới
  - Mô tả: Người dùng có thể tạo một công việc mới trong ứng dụng, nhập tiêu đề, mô tả và ngày hạn hoàn thành.
  - Ưu tiên: Must have (Phải có)
  - Story Points: 3

<b>Tên</b>	<b>Mô tả</b>	<b>Ưu tiên</b>	<b>Story point</b>
<b>Tạo công việc mới</b>	Người dùng có thể tạo một công việc mới, nhập tiêu đề, mô tả, mức độ ưu tiên	Must have	3
<b>Chỉnh sửa và cập nhật công việc</b>	Người dùng có thể chỉnh sửa thông tin công việc như tiêu đề, mô tả	Should have	3
<b>Đánh dấu công việc đã hoàn thành</b>	Người dùng có thể thay đổi trạng thái công việc sang “Đã hoàn thành”	Must have	2
<b>Xóa công việc</b>	Người dùng có thể xóa công việc khỏi danh sách	Must have	2
<b>Bộ lọc và tìm kiếm công việc</b>	Người dùng có thể lọc công việc theo trạng thái, mức độ ưu tiên hoặc tìm bằng từ khóa	Could have	5

4. Viết một Product Backlog cho một ứng dụng web cho phép người dùng tìm kiếm, chọn, và đặt vé cho các sự kiện trực tuyến hoặc trực tiếp.

Yêu cầu:

Liệt kê ít nhất 5 Product Backlog Items (PBI) cho các tính năng cần có trong ứng dụng.

Mỗi PBI phải có tên, mô tả ngắn gọn về chức năng, độ ưu tiên và ước lượng điểm Story Points (nếu có thể).

Product Backlog phải đảm bảo sự rõ ràng về các tính năng quan trọng cần được triển khai trong các Sprint tiếp theo.

Ví dụ:

1. Tìm kiếm sự kiện

- Mô tả: Người dùng có thể tìm kiếm sự kiện theo tên, loại sự kiện, và thời gian tổ chức.
- Ưu tiên: Must have (Phải có)
- Story Points: 3

Viết một Product Backlog cho một ứng dụng web cho phép người dùng tìm kiếm, chọn, và đặt vé cho các sự kiện trực tuyến hoặc trực tiếp.

Yêu cầu:

Liệt kê ít nhất 5 Product Backlog Items (PBI) cho các tính năng cần có trong ứng dụng.

Mỗi PBI phải có tên, mô tả ngắn gọn về chức năng, độ ưu tiên và ước lượng điểm Story Points (nếu có thể).

Product Backlog phải đảm bảo sự rõ ràng về các tính năng quan trọng cần được triển khai trong các Sprint tiếp theo.

Ví dụ:

1. Tìm kiếm sự kiện

- Mô tả: Người dùng có thể tìm kiếm sự kiện theo tên, loại sự kiện, và thời gian tổ chức.
- Ưu tiên: Must have (Phải có)
- Story Points: 3

Tên	Mô tả	Ưu tiên	Story point
Tìm kiếm sự kiện	Người dùng có thể tìm kiếm sự kiện theo tên, thể loại, địa điểm hoặc thời gian	Must have	3

<b>Xem chi tiết sự kiện</b>	Hiển thị thông tin chi tiết về sự kiện như mô tả, giá vé, địa điểm, ngày diễn ra, số lượng vé còn lại, hình ảnh	Must have	5
<b>Đặt vé sự kiện</b>	Người dùng chọn số lượng vé, loại vé và tiến hành đặt chỗ. Hệ thống kiểm tra số lượng vé còn lại trước khi xác nhận	Must have	8
<b>Thanh toán online</b>	Tích hợp phương thức thanh toán để người dùng thanh toán vé	Must have	8
<b>Đăng ký / Đăng nhập người dùng</b>	Người dùng tạo tài khoản, đăng nhập, quản lý thông tin cá nhân	Must have	5

5. Viết Acceptance Criteria cho User Story đã tạo ở bài tập 1 (Là một khách hàng, tôi muốn thêm sản phẩm vào giỏ hàng để có thể mua nhiều món trong cùng một đơn.). Cung cấp các tiêu chí chấp nhận rõ ràng và có thể kiểm thử được.

Yêu cầu:

- Viết ít nhất 3 tiêu chí chấp nhận (Acceptance Criteria) cho User Story.
- Mỗi tiêu chí chấp nhận cần phải rõ ràng, có thể kiểm thử được, và đảm bảo chất lượng tính năng theo yêu cầu của người dùng.

- ❖ Thêm sản phẩm vào giỏ hàng thành công
  - Given người dùng đang xem trang chi tiết sản phẩm
  - And sản phẩm còn hàng
  - When người dùng nhấn nút "Add to Cart"
  - Then hệ thống phải thêm sản phẩm vào giỏ hàng
  - And hiển thị thông báo xác nhận rằng sản phẩm đã được thêm.
- ❖ Giỏ hàng hiển thị đúng thông tin sản phẩm
  - Given người dùng đã thêm một sản phẩm vào giỏ hàng
  - When người dùng mở trang giỏ hàng
  - Then hệ thống phải hiển thị sản phẩm đã thêm
  - And hiển thị đầy đủ thông tin (tên, giá, số lượng, hình ảnh).
- ❖ Thêm cùng sản phẩm nhiều lần sẽ tăng số lượng
  - Given người dùng đã có sản phẩm trong giỏ hàng
  - When người dùng nhấn "Add to Cart" cùng sản phẩm đó lần nữa
  - Then số lượng của sản phẩm đó trong giỏ hàng phải tăng lên 1
  - And tổng giá trị giỏ hàng phải được cập nhật.

6. Dựa trên Product Backlog bạn đã tạo ở bài tập 2 (ứng dụng quản lý công việc), hãy phân loại các yêu cầu theo phương pháp MoSCoW (Must have, Should have, Could have, Won't have).

Yêu cầu:

- Phân loại ít nhất 5 mục trong Product Backlog của bạn theo 4 mức độ ưu tiên:
  - Must have (Phải có): Những yêu cầu quan trọng nhất, không thể thiếu để sản phẩm hoạt động đúng chức năng.
  - Should have (Nên có): Các yêu cầu có thể trì hoãn một chút nhưng mang lại giá trị cao cho người dùng.
  - Could have (Có thể có): Những tính năng cải tiến nhỏ, không ảnh hưởng đến chức năng chính.

- Won't have (Không cần có): Các tính năng không mang lại giá trị lớn và có thể hoãn hoặc loại bỏ trong Sprint hiện tại.

❖ Thêm sản phẩm vào giỏ hàng thành công

- Given người dùng đang xem trang chi tiết sản phẩm
- And sản phẩm còn hàng
- When người dùng nhấn nút "Add to Cart"
- Then hệ thống phải thêm sản phẩm vào giỏ hàng
- And hiển thị thông báo xác nhận rằng sản phẩm đã được thêm.

❖ Giỏ hàng hiển thị đúng thông tin sản phẩm

- Given người dùng đã thêm một sản phẩm vào giỏ hàng
- When người dùng mở trang giỏ hàng
- Then hệ thống phải hiển thị sản phẩm đã thêm
- And hiển thị đầy đủ thông tin (tên, giá, số lượng, hình ảnh).

❖ Thêm cùng sản phẩm nhiều lần sẽ tăng số lượng

- Given người dùng đã có sản phẩm trong giỏ hàng
- When người dùng nhấn "Add to Cart" cùng sản phẩm đó lần nữa
- Then số lượng của sản phẩm đó trong giỏ hàng phải tăng lên 1
- And tổng giá trị giỏ hàng phải được cập nhật.

7. Sử dụng User Story của bài tập 1 (một ứng dụng bán hàng trực tuyến, trong đó người dùng có thể thêm sản phẩm vào giỏ hàng) và viết lại nó với các Acceptance Criteria rõ ràng, sử dụng phương pháp Gherkin (Given – When – Then).

Yêu cầu:

- Viết lại User Story theo cấu trúc Gherkin và thêm ít nhất 3 Acceptance Criteria.
  - Cấu trúc Gherkin:
    - Given: Điều kiện ban đầu (trạng thái ban đầu của hệ thống).
    - When: Hành động mà người dùng thực hiện.

■ Then: Kết quả mong đợi sau hành động.

- ❖ Thêm sản phẩm vào giỏ hàng thành công
  - Given người dùng đang xem trang chi tiết sản phẩm
  - And sản phẩm còn hàng
  - When người dùng nhấn nút "Add to Cart"
  - Then hệ thống phải thêm sản phẩm vào giỏ hàng
  - And hiển thị thông báo xác nhận rằng sản phẩm đã được thêm.
- ❖ Giỏ hàng hiển thị đúng thông tin sản phẩm
  - Given người dùng đã thêm một sản phẩm vào giỏ hàng
  - When người dùng mở trang giỏ hàng
  - Then hệ thống phải hiển thị sản phẩm đã thêm
  - And hiển thị đầy đủ thông tin (tên, giá, số lượng, hình ảnh).
- ❖ Thêm cùng sản phẩm nhiều lần sẽ tăng số lượng
  - Given người dùng đã có sản phẩm trong giỏ hàng
  - When người dùng nhấn "Add to Cart" cùng sản phẩm đó lần nữa
  - Then số lượng của sản phẩm đó trong giỏ hàng phải tăng lên 1
  - And tổng giá trị giỏ hàng phải được cập nhật.

8. Dựa trên User Story và Acceptance Criteria đã tạo ở bài tập 7, xác định một Definition of Done cho User Story đó.

Yêu cầu:

- Đưa ra ít nhất 5 tiêu chuẩn trong Definition of Done.
  - Definition of Done (DoD): Là bộ tiêu chuẩn chung mà tất cả các User Stories hoặc sản phẩm phải đáp ứng trước khi được coi là "Done".



- Các tiêu chuẩn này đảm bảo chất lượng của sản phẩm cuối cùng và giúp tất cả các bên liên quan hiểu rõ khi nào công việc được coi là hoàn thành.
- ❖ Tất cả Acceptance Criteria đều đúng
  - Các tình huống Gherkin đều đã được kiểm thử và hoạt động chính xác.
- ❖ Không còn lỗi
  - Chức năng thêm sản phẩm vào giỏ hàng chạy đúng, không bị lỗi nghiêm trọng.
- ❖ Giao diện hiển thị chính xác
  - Nút “Add to Cart”, thông báo thêm sản phẩm và giỏ hàng đều hiển thị đúng và dễ sử dụng.
- ❖ Dữ liệu được lưu đúng
  - Sản phẩm được thêm vào giỏ hàng chính xác, số lượng cập nhật đúng khi thêm nhiều lần.
- ❖ Code đã được kiểm tra và chấp nhận
  - Code đã được đồng đội kiểm tra và được đưa vào phiên bản chính của dự án.