Progetto: Piattaforma E-Commerce “Artigianato Online”

1. Descrizione del Caso d’Uso

Scenario:

Il cliente è una startup che vuole vendere online prodotti artigianali realizzati da creativi locali. La piattaforma consente agli artigiani di creare un profilo, caricare i propri prodotti, gestire l’inventario e gli ordini, mentre i clienti possono sfogliare il catalogo, fare acquisti e gestire il proprio profilo.

Obiettivi principali:

- **Catalogo prodotti:** Visualizzazione dei prodotti con filtri per categoria, prezzo e disponibilità.
- **Gestione account:** Registrazione e accesso per artigiani e clienti.
- **Processo d’ordine:** Acquisto, pagamento e gestione degli ordini.
- **Pannello di controllo:** Dashboard per gli artigiani per monitorare vendite, aggiornare prodotti e gestire lo stock.
- **Deployment Cloud:** L’applicazione finale deve essere deployata su una piattaforma cloud (non in locale), garantendo scalabilità ed accessibilità globale.

---

2. Intervista Simulata col Potenziale Cliente

Intervistatore:
“Buongiorno, ci parli un po’ della vostra idea per la piattaforma di e-commerce. Quali sono gli obiettivi principali?”

Cliente (Artigianato Online):
“Vogliamo creare un punto di incontro tra artigiani e clienti. Gli artigiani devono poter esporre i propri prodotti e gestire facilmente ordini e inventario. I clienti, invece, devono avere un’interfaccia intuitiva per trovare e acquistare prodotti in maniera sicura, con opzioni di pagamento multiple.”

Intervistatore:
“Chi sono gli utenti che utilizzeranno il sistema?”

Cliente:
“Principali utenti:

1. **Artigiani:** Creatori che caricano e gestiscono i propri prodotti.

2. **Clienti finali:** Consumatori che navigano e acquistano i prodotti.

3. **Amministratore:** Un operatore che monitora l’attività della piattaforma, gestisce eventuali segnalazioni e garantisce il corretto funzionamento generale.”

Intervistatore:
“Quali funzionalità ritenete indispensabili e quali aspetti tecnici sono critici?”

Cliente:
“È fondamentale una ricerca efficace nel catalogo e un checkout veloce. La sicurezza è prioritaria, sia per i dati degli utenti sia per i pagamenti. Inoltre, vogliamo scalabilità: il sistema dovrà essere deployato su cloud per gestire facilmente picchi di traffico e crescere nel tempo.”

Intervistatore:
“Ci parli del deploy: avete preferenze sulla piattaforma cloud?”

Cliente:
“Siamo aperti a soluzioni come AWS, Google Cloud o Heroku, purché il team possa scegliere quella più adatta in base alle proprie competenze. Il deploy su cloud è obbligatorio; non deve rimanere in ambiente locale.”

3. Divisione dei Ruoli e Tempistiche

Ruoli del Team

Frontend Developer(s)

- **Responsabilità:**
    - Progettare e sviluppare l’interfaccia utente responsive per la visualizzazione del catalogo, gestione del carrello e pagine utente.
    - Implementare il checkout e la comunicazione dinamica con il backend mediante API.

- **Deliverables:**
    - Codice HTML/CSS/JS e prototipi UI, con interazioni fluide e responsive.
    - Documentazione dei componenti grafici e test di compatibilità browser.

Backend Developer(s)

- **Responsabilità:**
    - Realizzare l'API RESTful in Node.js/Express per la gestione di utenti, prodotti, ordini e pagamenti.
    - Implementare logica di sicurezza e validazione dei dati, incluso l'integrazione per i pagamenti.
- **Deliverables:**
    - Codice del backend, documentazione degli endpoint (es. Swagger/OpenAPI).
    - Implementazione di test unitari e integration test.

Database Developer/Specialista

- **Responsabilità:**
    - Progettare lo schema relazionale per gestire utenti, prodotti, ordini e transazioni.
    - Creare script SQL per la configurazione, popolamento iniziale e backup del database.
- **Deliverables:**
    - Diagrammi ER, script SQL e documentazione sulle operazioni CRUD.
    - Guide per la migrazione e gestione del database in ambiente cloud.

DevOps/Infrastruttura

- **Responsabilità:**

    - Configurare l’ambiente cloud scelto per il deploy dell’applicazione.
    - Gestire l’integrazione continua e il monitoraggio dell’applicazione in produzione.
- **Deliverables:**
    - Script di deployment (es. Dockerfile, CI/CD pipeline).
    - Documentazione dettagliata per il deploy, scaling e monitoraggio in cloud.

4. Modalità di Consegna

- **Repository Git:**
    - Strutturato in branch specifici per frontend, backend, script del database e setup di deployment.
    - README esaustivo con istruzioni per la configurazione e il deploy in cloud, includendo riferimenti alla documentazione.
- **Documentazione Tecnica:**
    - Specifiche degli endpoint API, diagrammi ER, guide per il deploy su cloud e istruzioni dettagliate per il rollback in caso di problemi.
- **Demo:**
    - Dimostrazione finale tramite sessione live o video che mostri l’applicazione funzionante in ambiente cloud.
- **Testing:**
    - Integrazione di test automatici e report che documentino il corretto funzionamento e la scalabilità dell’applicazione.

---

5. Criteri di Valutazione

- **Funzionalità:**

    - Completezza delle funzionalità per la gestione del catalogo, account, ordine e dashboard.

- **Qualità del Codice:**
    - Adesione agli standard di codifica e uso corretto dei design pattern.
- **UI/UX:**
    - Interfaccia utente intuitiva, responsive e adatta a dispositivi desktop e mobile.
- **Integrazione e Sicurezza:**
    - Efficace coordinazione tra frontend, backend e database, con particolare attenzione a sicurezza e protezione dei dati.
- **Deploy e Scalabilità:**
    - Successo del deploy su ambiente cloud, con documentazione chiara e test di performance.
- **Documentazione e Testing:**
    - Completezza e chiarezza della documentazione, inclusi i report di test