**Punti svolti Punti da svolgere Punti da tenere in considerazione**

**Per superare la discussione del progetto devono essere seguite nel dettaglio le seguenti linee guida. Il progetto è di gruppo (max quattro persone). E' possibile suddividersi i compiti durante lo sviluppo, ma nel momento della discussione tutti devono essere a conoscenza dei contenuti e delle funzionalità dell'intero progetto. La Base di Dati ha un ruolo molto importante.**

Il sito deve essere di commercio elettronico:

- il cliente deve poter inserire prodotti nel carrello, variarne la quantità.

Una volta confermato l'ordine deve essere possibile visualizzare l'ordine nell'elenco degli ordini effettuati e va svuotato il carrello.

Va prevista la figura dell'amministratore e delle pagine a lui dedicate, accessibili solo dopo autenticazione (vedere lezione su Security). Usare autenticazione programmata.

L'AMMINISTRATORE DEVE POTER INSERIRE, MODIFICARE, VISUALIZZARE E CANCELLARE ELEMENTI DEL CATALOGO, VISUALIZZARE GLI ORDINI COMPLESSIVI, DALLA DATA ALLA DATA E PER CLIENTE.

**REQUISITO MOLTO IMPORTANTE: IL DATABASE DEVE ESSERE STRUTTURATO IN MODO TALE CHE SE VENGONO MODIFICATI IL PREZZO O L'IVA DI UN PRODOTTO ACQUISTATO SUCCESSIVAMENTE ALL'ACQUISTO L'ORDINE DEL CLIENTE MANTIENE I DATI CORRETTI.**

Va mantenuto il vincolo d'**integrità referenziale**: se l'amministratore cancella un prodotto non deve scomparire dagli ordini effettuati.

Prevenire SQL injection

**usare i filtri +++ NEW +++**

·     Il sito deve essere responsive;

·     Il sito deve girare su Tomcat direttamente;

·     Usare il modello [MVC](http://elearning.informatica.unisa.it/el-platform/mod/assign/view.php?id=10410);

·     Creare almeno due package: uno per le servlet, chiamato Control, ed uno per il Model, chiamato Model.

·     Il Model deve contenere i bean, il carrello;

·     Il codice HTML viene creato esclusivamente dalle JSP. JSP e HTML che formano il view

.    Usare il datasource o drivemanager per connettersi al DB (esempio storage). Se si usa DriveManager utilizzare anche Connection Pool.

·     I form sono controllati con javascript. Il form viene inviato al server solo se corretto. Usare le espressioni regolari per validare i campi del form. Mettere il "focus" sul campo in cui l'utente sta scrivendo. Visualizzare le istruzioni di compilazione di ogni campo di input nel placeholder. Fornire i messaggi di errore quando l'utente preme il submit (evitare gli alert).

·

·     Usare AJAX per scambiare piccole informazioni con il server (in formato JSON)

       Almeno: utilizzare ajax per la barra di ricerca (come in google suggest) e controllare durante la registrazione che l'email non è già presente nel database

·     Gestire le sessioni per memorizzare il carrello. Si salva l'ordine nel DB dopo l'acquisto.

DARE conferma all'utente - registrazione effettuata con successo, prodotto inserito con successo"

quando amministratore cancella chiedere conferma prima di eseguie

**Gestire pagine di errore. il server non deve perdere il controllo in caso di errore**

**(risorsa non trovata, server indisponibile, mancanza di permessi per l'accesso)**

.   Usare i fragment (con include) nelle pagine JSP per creare header, footer e menu (esempio progetto negozio, L09 bis, JSP).