|  |  |
| --- | --- |
| **TEMPLATE DI PROJECT WORK = 3 CFU**  **min 12 pagine - max 20 pagine**  *\*da compilare e caricare in formato pdf* | |
| **Cognome e Nome:** | MUTHUKUDA ARACHCHIGE THILINA SARIDU |
| **Numero di Matricola**: | 0312301683 |
| **Corso di Studio:** | Barrare la casella riferita al proprio corso di studio |
| ◊   L-5 Filosofia ed Etica |
| ◊   L-22 Scienze Motorie |
| ◊   L-31 Informatica per le Aziende Digitali |
|  |
| **Tema n:** | 1 |
| **Titolo del tema:** | La digitalizzazione dell’impresa |
| **Traccia del PW n:** | 1,4 |
| **Titolo della traccia:** | Sviluppo di una pagina web per un servizio di prenotazione online di un’impresa del settore terziario |
| **Titolo dell’elaborato:** | Sistema di Prenotazione Online per un Ristorante |
| **PARTE PRIMA – DESCRIZIONE DEL PROCESSO** | |
| **Utilizzo delle conoscenze e abilità derivate dal percorso di studio:** | ****Conoscenze e Abilità Derivate dal Percorso di Studio**** Durante la realizzazione dell’elaborato, sono state applicate competenze provenienti dai seguenti insegnamenti del tuo corso di laurea in Informatica per le Aziende Digitali. Di seguito, per ciascun insegnamento pertinente, descrivo il contributo specifico: ****1. Programmazione 1 (INF/01)**** Questo insegnamento ha fornito le basi della programmazione, fondamentali per strutturare il progetto. Ha contribuito a comprendere i principi fondamentali di sviluppo software, come la gestione dei dati, l’implementazione di algoritmi e l’ottimizzazione del codice.   * **Contributo al progetto:**   + Scrittura del codice backend in Python.   + Gestione dei cicli, condizioni e funzioni per implementare le logiche di prenotazione.   + Applicazione di principi di programmazione strutturata e modulare.  ****2. Basi di Dati (ING-INF/05)**** Questo insegnamento ha fornito conoscenze sulla modellazione e gestione di database relazionali. È stato centrale per progettare e gestire il database SQLite del progetto.   * **Contributo al progetto:**   + Creazione della tabella bookings per archiviare i dati delle prenotazioni.   + Scrittura di query SQL per inserire e recuperare informazioni.   + Gestione della persistenza dei dati nel sistema di prenotazione.  ****3. Algoritmi e Strutture Dati (INF/01)**** Questo corso ha rafforzato la capacità di analizzare e ottimizzare le logiche applicative, con un focus su efficienza e correttezza degli algoritmi.   * **Contributo al progetto:**   + Implementazione di logiche efficienti nel backend per elaborare i dati delle prenotazioni.   + Progettazione di algoritmi per garantire la consistenza dei dati e la gestione delle eccezioni.  ****4. Ingegneria del Software (ING-INF/05)**** Questo insegnamento ha introdotto le metodologie di progettazione e sviluppo software, fondamentali per gestire un progetto complesso come il sistema di prenotazione.   * **Contributo al progetto:**   + Pianificazione del progetto in fasi: analisi, progettazione, sviluppo e test.   + Uso di best practice per garantire un codice robusto e leggibile.   + Implementazione del sistema con un approccio modulare, separando le responsabilità tra frontend e backend.  ****5. Tecnologie Web (INF/01)**** Questo insegnamento ha fornito le competenze per sviluppare applicazioni web utilizzando HTML, CSS e JavaScript, strumenti fondamentali per il frontend del progetto.   * **Contributo al progetto:**   + Creazione delle pagine web per la visualizzazione del menu e del modulo di prenotazione.   + Applicazione di CSS per il design responsivo e di JavaScript per le interazioni dinamiche con l’utente.   + Gestione delle chiamate API al backend utilizzando fetch API.  ****6. Architettura dei Calcolatori (INF/01)**** L’insegnamento ha contribuito alla comprensione del funzionamento dei sistemi informatici, fornendo una base per sviluppare software efficiente.   * **Contributo al progetto:**   + Comprensione delle interazioni tra le componenti del sistema, come server, database e client.   + Ottimizzazione delle risorse computazionali durante l’elaborazione delle richieste API.  ****7. Reti di Calcolatori e Cybersecurity (INF/01)**** Questo insegnamento ha fornito conoscenze sulla comunicazione tra sistemi e sulla sicurezza informatica.   * **Contributo al progetto:**   + Implementazione della comunicazione tra frontend e backend utilizzando HTTP.   + Applicazione di misure base di sicurezza per proteggere i dati degli utenti, come la validazione e l’autenticazione delle richieste.  ****8. Strategia, Organizzazione e Marketing (ING-IND/35)**** Questo corso ha offerto una visione aziendale che ha guidato la progettazione del sistema per soddisfare le esigenze reali del ristorante.   * **Contributo al progetto:**   + Comprensione delle esigenze operative e di marketing del ristorante.   + Creazione di una piattaforma user-friendly per migliorare l’esperienza cliente e supportare le operazioni aziendali.  ****9. Logica (INF/01)**** Questo insegnamento ha migliorato le capacità di ragionamento logico, fondamentali per scrivere un codice corretto e privo di errori.   * **Contributo al progetto:**   + Progettazione di condizioni logiche per validare i dati inseriti dagli utenti.   + Strutturazione di algoritmi che garantiscano la consistenza dei dati nel sistema.  ****Conclusioni**** La redazione dell’elaborato ha rappresentato un’opportunità per applicare in modo concreto le competenze teoriche e pratiche acquisite durante il percorso di studi. Ogni insegnamento ha contribuito in modo significativo a sviluppare una comprensione approfondita e integrata delle tecnologie informatiche, dimostrando l’importanza di un approccio interdisciplinare per affrontare sfide reali. Questo progetto non solo consolida quanto appreso, ma offre anche una base solida per affrontare progetti futuri in ambito professionale. |
| **Fasi di lavoro e relativi tempi di implementazione per la predisposizione dell’elaborato:** | ****Fasi di lavoro e relativi tempi di implementazione per la predisposizione dell’elaborato**** La redazione dell’elaborato è stata organizzata in diverse fasi, ciascuna con obiettivi specifici e tempi ben definiti. Ogni fase è stata affrontata seguendo un approccio metodico, che ha incluso analisi, progettazione, implementazione e verifica. Di seguito, sono descritte in dettaglio le attività svolte, il tempo dedicato a ciascuna fase, le difficoltà incontrate e le soluzioni adottate per superarle. ****1. Analisi delle Specifiche (5 giorni)**** **Attività svolte**:   * Studio della traccia del progetto per identificare le funzionalità richieste. * Analisi delle esigenze del ristorante "La Buona Tavola", come la gestione delle prenotazioni e la necessità di una piattaforma user-friendly. * Definizione degli obiettivi del progetto: creazione di una home page informativa, un modulo di prenotazione e un sistema backend per la gestione delle richieste.   **Difficoltà incontrate**:   * Definire con precisione le funzionalità prioritarie per il sistema, bilanciando semplicità e completezza. * Interpretare i requisiti del progetto in termini pratici.   **Soluzioni adottate**:   * Creazione di una lista di funzionalità essenziali (ad esempio: gestione delle prenotazioni) e opzionali (ad esempio: notifiche email) per organizzare il lavoro in fasi.  ****2. Progettazione del Sistema (7 giorni)**** **Attività svolte**:   * Creazione di wireframe per definire il layout delle pagine web (home page e modulo di prenotazione). * Progettazione della struttura del database SQLite, con una tabella bookings per memorizzare i dati delle prenotazioni. * Definizione delle rotte API per il backend Flask, inclusa la gestione delle richieste POST per l’inserimento dei dati.   **Difficoltà incontrate**:   * Decidere come organizzare i dati nel database per garantire flessibilità e sicurezza. * Progettare un’interfaccia utente semplice ma esteticamente gradevole.   **Soluzioni adottate**:   * Utilizzo di strumenti di progettazione come Figma per creare mockup visivi. * Consultazione della documentazione SQLite per progettare una struttura dati ottimale. * Studio delle best practices di design per siti di prenotazione online.  ****3. Implementazione del Frontend (10 giorni)**** **Attività svolte**:   * Creazione della pagina home (index.html) con sezioni informative e immagini dei piatti principali. * Sviluppo del modulo di prenotazione (booking.html) con campi per nome, email, telefono, data, orario e numero di ospiti. * Applicazione di CSS per garantire un design responsivo e un layout moderno. * Implementazione di JavaScript per la validazione dei dati e l’interazione con l’utente (ad esempio: popup di conferma).   **Difficoltà incontrate**:   * Garantire la compatibilità del design su dispositivi di dimensioni diverse. * Gestire le chiamate API dal frontend al backend senza causare errori.   **Soluzioni adottate**:   * Test iterativi su diversi browser e dispositivi per assicurare la responsività. * Debugging del codice JavaScript utilizzando strumenti del browser (console) per identificare e risolvere eventuali problemi.  ****4. Implementazione del Backend (7 giorni)**** **Attività svolte**:   * Configurazione di Flask per gestire le richieste API. * Creazione della rotta /api/book per ricevere i dati delle prenotazioni e salvarli nel database. * Implementazione della gestione degli errori nel backend per garantire l’affidabilità del sistema.   **Difficoltà incontrate**:   * Sincronizzazione dei dati tra frontend e backend. * Gestione delle eccezioni in caso di input non validi o errori di connessione al database.   **Soluzioni adottate**:   * Utilizzo di Postman per testare le chiamate API e verificare il corretto funzionamento del backend. * Implementazione di controlli di validazione lato server per garantire l’integrità dei dati.  ****5. Test e Debugging (2 giorni)**** **Attività svolte**:   * Verifica delle funzionalità dell’intero sistema (frontend, backend e database). * Simulazione di scenari reali per testare l’inserimento, la gestione e la visualizzazione delle prenotazioni. * Risoluzione di bug e miglioramento delle prestazioni complessive.   **Difficoltà incontrate**:   * Individuare errori nelle comunicazioni tra il modulo di prenotazione e il backend. * Garantire la gestione corretta delle eccezioni per evitare crash del sistema.   **Soluzioni adottate**:   * Debugging approfondito utilizzando log dettagliati in Flask e strumenti di sviluppo del browser. * Test di stress per simulare un utilizzo intensivo e verificare la stabilità del sistema.  ****6. Documentazione del Progetto (4 giorni)**** **Attività svolte**:   * Redazione di un rapporto dettagliato che descrive il contesto, le specifiche e le fasi di sviluppo del progetto. * Inclusione del codice sorgente con commenti dettagliati per facilitare la comprensione delle scelte progettuali.   **Difficoltà incontrate**:   * Organizzare le informazioni in modo chiaro e comprensibile, rispettando il formato richiesto. * Assicurarsi che tutti gli aspetti del progetto siano adeguatamente documentati.   **Soluzioni adottate**:   * Utilizzo di un formato strutturato per il rapporto, suddiviso in sezioni chiare. * Revisione iterativa del documento per garantire completezza e accuratezza.  ****Conclusioni:**** Il progetto è stato completato con successo, rispettando i tempi pianificati per ciascuna fase. Le difficoltà incontrate, come la gestione delle comunicazioni API e l’ottimizzazione del design responsivo, sono state superate grazie a un approccio iterativo e all’utilizzo di strumenti di debugging e test. Questa esperienza ha consolidato le competenze acquisite durante il percorso di studi e ha dimostrato  l’importanza di una pianificazione accurata per affrontare un progetto complesso. |
|  |
|  |
|  |
|  |
| **Risorse e strumenti impiegati:** | Risorse e strumenti individuati e utilizzati: La redazione dell’elaborato ha richiesto l’utilizzo di risorse teoriche e pratiche accuratamente selezionate. Sono stati consultati:   * La documentazione ufficiale di **Flask** per la creazione delle API e la gestione delle rotte web; * Le guide ufficiali di **SQLite** per l’implementazione del database relazionale leggero; * Materiale accademico, tra cui dispense e appunti di corsi di sviluppo web, basi di dati e programmazione distribuita, per solidificare i fondamenti teorici; * Community tecniche online (Stack Overflow, forum Flask) per risolvere problematiche specifiche.   Gli strumenti pratici utilizzati comprendono:   * **Visual Studio Code** come ambiente di sviluppo integrato per la scrittura e gestione del codice; * **Postman** e curl per il testing delle API REST; * **Browser Developer Tools** (console) per monitorare errori di front-end; * Librerie **HTML, CSS e JavaScript** per il front-end e **Python/Flask** per il back-end. |
| Motivazioni della scelta delle risorse e degli strumenti: La scelta delle risorse si è basata su criteri di affidabilità, aggiornamento e ampia diffusione. La documentazione ufficiale di Flask e SQLite è stata preferita in quanto fornisce esempi chiari e procedure aggiornate rispetto alle ultime versioni dei software. Visual Studio Code è stato scelto per la sua leggerezza e la sua capacità di supportare contemporaneamente più linguaggi (Python, HTML, CSS, JavaScript), facilitando l'integrazione tra le diverse componenti del progetto.  Postman e curl sono stati selezionati per la loro efficacia nel testare rapidamente le API e per simulare scenari realistici di interazione utente. La scelta delle tecnologie open-source, infine, risponde sia a criteri economici sia alla volontà di adottare strumenti standard nel settore dello sviluppo web, garantendo compatibilità e scalabilità future. |
| Modalità di individuazione e reperimento delle risorse e degli strumenti: Le risorse principali sono state reperite attraverso la **consultazione diretta dei siti ufficiali** delle tecnologie utilizzate, in particolare:   * flask.palletsprojects.com per Flask; * sqlite.org per SQLite; * code.visualstudio.com per Visual Studio Code.   In aggiunta, per la risoluzione di problemi specifici e dubbi durante lo sviluppo, è stato utile fare riferimento a forum tecnici e community online come **Stack Overflow**, dove è stato possibile trovare risposte dettagliate a problematiche comuni.  La scelta delle estensioni per Visual Studio Code e delle librerie front-end è avvenuta testando varie soluzioni disponibili e selezionando quelle più adatte alle esigenze del progetto, privilegiando strumenti stabili e ben documentati. |
| Difficoltà affrontate e soluzioni adottate: Tra le principali difficoltà affrontate vi è stata la gestione delle sessioni utente, necessaria per proteggere le rotte riservate al ristoratore e agli utenti finali. La configurazione delle sessioni e la loro corretta implementazione in Flask hanno richiesto una comprensione approfondita delle modalità di autenticazione e della protezione degli endpoint, superata tramite lettura attenta della documentazione ufficiale e prove iterative.  Un'altra sfida si è presentata nella realizzazione delle funzionalità di modifica e cancellazione delle prenotazioni, che ha implicato la creazione di nuove rotte API e l'integrazione con il front-end per gestire dinamicamente le operazioni. La sincronizzazione tra front-end e back-end, specie nella gestione degli stati delle richieste e nella visualizzazione dinamica dei dati, ha richiesto debug accurati e l’uso sistematico di strumenti come Postman e la console del browser per tracciare eventuali anomalie.  Infine, una sfida tecnica è stata la configurazione degli slot orari di prenotazione per limitarli a intervalli di 30 minuti. Questo ha richiesto la modifica del form HTML e un’attenta gestione dei dati inviati al server, con il fine di prevenire errori di validazione e garantire un’esperienza utente coerente.  Tutte le difficoltà sono state affrontate con un approccio iterativo, che ha previsto la realizzazione di versioni incrementali del sistema e test approfonditi ad ogni nuova funzionalità implementata. |
| **PARTE SECONDA – PREDISPOSIZIONE DELL’ELABORATO** | |
| **Obiettivi dell’elaborato/progetto/artefatto:** | L’obiettivo principale dell’elaborato è stato lo sviluppo di un **sistema di prenotazione online** per un ristorante, conforme alle specifiche indicate nella traccia del Project Work. Il progetto ha raggiunto i seguenti risultati:   * Creazione di una **pagina web responsive e intuitiva**, realizzata con HTML, CSS e JavaScript, che consente agli utenti di visualizzare informazioni generali sul ristorante, il menu e accedere alla funzionalità di prenotazione; * Implementazione di un **modulo di prenotazione online** che permette all’utente di inserire i propri dati (nome, email, telefono), la data e l’orario della prenotazione, scegliendo tra slot di 30 minuti predefiniti; * Sviluppo di un **back-end in Python/Flask**, con API REST che gestiscono l’invio, la memorizzazione e il recupero delle prenotazioni in un database SQLite; * Introduzione di una **dashboard riservata al ristoratore** per la gestione delle prenotazioni, con possibilità di visualizzare, modificare ed eliminare le prenotazioni esistenti; * Implementazione di un **sistema di autenticazione** per proteggere le funzionalità riservate sia per il ristoratore sia per gli utenti, che possono verificare e modificare la propria prenotazione dopo aver effettuato l’accesso.   Questo progetto risponde pienamente alle richieste della traccia, non solo realizzando la parte tecnica ma ampliando le funzionalità con elementi di **gestione autonoma delle prenotazioni** da parte degli utenti e del ristoratore, incrementando così l’usabilità e l’efficienza del sistema. |
| **Contestualizzazione:** | L’elaborato si inserisce nell’ambito teorico-applicativo della **digitalizzazione dei servizi per il settore terziario**, un tema di crescente rilevanza nel panorama economico e sociale attuale. In particolare, il ristorante scelto come caso d’uso rappresenta un esempio tipico di piccola-media impresa che necessita di strumenti tecnologici per ottimizzare la gestione delle prenotazioni e migliorare l’esperienza del cliente.  La pandemia e i mutamenti nel comportamento dei consumatori hanno accelerato la richiesta di **soluzioni digitali accessibili e sicure** per la gestione delle prenotazioni e degli ordini online. In questo contesto, il progetto sviluppato dimostra come le competenze di programmazione web, gestione di database e sicurezza delle applicazioni possano tradursi in strumenti concreti e funzionali, favorendo la **modernizzazione dei processi aziendali**.  Dal punto di vista teorico, l’elaborato integra concetti di **architettura client-server, sviluppo full-stack** e sicurezza web, creando un ecosistema applicativo bilanciato tra front-end e back-end, coerente con le best practices del settore. |
| **Descrizione dei principali aspetti progettuali:** | Sviluppare l’elaborato richiesto dalla traccia prescelta. |
| **Campi di applicazione:** | L’applicazione progettata trova la sua **applicazione principale** nel settore della ristorazione, ma il suo schema architetturale è facilmente adattabile a molteplici ambiti del settore terziario che necessitano di un sistema di prenotazione, tra cui:   * **Hotel e strutture ricettive** per la gestione delle camere; * **Servizi di trasporto privato** (taxi, noleggio auto); * **Centri estetici e parrucchieri** per la gestione degli appuntamenti; * **Spazi di coworking** o sale conferenze; * **Centri sportivi** per la prenotazione di campi da gioco o corsi.   I vantaggi principali derivanti dall'applicazione di questo sistema includono:   * **Automazione del processo di prenotazione**, che riduce il rischio di errori manuali; * **Maggiore comodità per l’utente finale**, che può gestire in autonomia la propria prenotazione; * **Efficienza nella gestione aziendale**, grazie alla disponibilità immediata dei dati e alla possibilità di monitorare facilmente l’andamento delle prenotazioni; * **Flessibilità**: l’architettura modulare del sistema consente personalizzazioni rapide e adattamenti a diversi contesti aziendali. |
| **Valutazione dei risultati (potenzialità e criticità):** | L’elaborato ha dimostrato una **buona efficacia pratica**, raggiungendo pienamente gli obiettivi prefissati. La **potenzialità principale** del sistema è la sua **scalabilità**: può essere ampliato facilmente per includere funzionalità più complesse, come notifiche email automatiche, sistemi di pagamento online o dashboard avanzate per analizzare le prenotazioni nel tempo.  Un altro punto di forza è la **semplicità d’uso**, sia per l’utente finale sia per il ristoratore, che consente di integrare l’applicazione rapidamente senza la necessità di formazione tecnica specifica.  Tra le **criticità riscontrate**, si evidenziano:   * La **sicurezza**: pur essendo implementata un’autenticazione di base, per un utilizzo reale su larga scala sarebbe necessario implementare sistemi più avanzati, come la crittografia delle password e la protezione da attacchi di tipo CSRF e XSS; * La **gestione della concorrenza**: attualmente, il sistema non gestisce casi di overbooking in tempo reale. Se due utenti prenotano lo stesso slot contemporaneamente, il sistema accetta entrambe le prenotazioni senza bloccare conflitti; * La **mancanza di notifiche automatiche**: al momento, la conferma di prenotazione è solo simulata, senza invio reale di email.   Questi limiti non compromettono la funzionalità del prototipo, ma rappresentano spunti di miglioramento futuri per trasformare il sistema in una soluzione completamente production-ready. |