1. Informazioni generali del PRD

Titolo del progetto: Web App “PromessiSposi.io” Visione di prodotto: Offrire un’esperienza didattica immersiva e ludica per conoscere e memorizzare la trama, i personaggi e i temi de I Promessi Sposi, incentivando la partecipazione quotidiana attraverso meccaniche di gamification, progressione a livelli e ricompense digitali. Data di redazione: 4 giugno 2025 Stakeholder principali: Product Manager (definisce requisiti e roadmap) Team di sviluppo (frontend, backend, database) Team UX/UI design Team contenuti (esperti di letteratura italiana) Marketing e Growth Utenti finali (studenti, appassionati, professori)

2. Obiettivi di business e metriche di successo Engagement giornaliero Obiettivo iniziale: almeno il 30% degli utenti attivi torna quotidianamente. Misurazione: calcolo del rapporto DAU/MAU con target ≥ 20%. Tasso di completamento lezioni Obiettivo: almeno il 60% degli utenti completa ogni unità formativa entro 48 ore dall’apertura. Misurazione: percentuale di lezioni concluse entro 48 ore. Retention a 30 giorni Obiettivo: il 20% degli utenti rimane attivo dopo 30 giorni dal primo accesso. Misurazione: analisi delle coorti utente a 30 giorni. Virality e referral Obiettivo: generare almeno 500 inviti utente (referral) entro 6 mesi dal lancio. Misurazione: conteggio dei codici referral utilizzati e nuovi account creati tramite referral. Revenue target (modello freemium) Obiettivo a 12 mesi: € 5.000 mensili di entrate ricorrenti da abbonamenti Premium e acquisti in-app. Misurazione: MRR (Monthly Recurring Revenue) e ARPU (Average Revenue per User).

3. Descrizione degli utenti target Studenti delle scuole superiori (14–19 anni) Motivazione: ripasso per prove d’esame di italiano, approfondimento scolastico. Bisogni: supporto alla comprensione del testo, glossario intuitivo, esercizi interattivi. Appassionati di letteratura (20–60 anni) Motivazione: voglia di approfondire la conoscenza de I Promessi Sposi in modo interattivo e coinvolgente. Bisogni: contenuti storici e culturali di contesto, modalità di esplorazione autonoma. Professori di italiano Motivazione: utilizzare la piattaforma come strumento didattico in classe o assegnare compiti a casa. Bisogni: monitoraggio progressi degli studenti, esercizi calibrati, risorse aggiuntive (es. mappe, note critiche).

4. Requisiti funzionali principali Struttura delle unità didattiche Suddivisione in moduli corrispondenti a capitoli o sezioni tematiche. Ogni modulo comprende: Lettura interattiva: testo del capitolo con note contestuali (tooltip) e glossario a comparsa per termini arcaici o complessi. Sidebar “Approfondisci”: Spiegazioni storiche (es. contesto sociale, ambientazione lombarda del XVII secolo). Approfondimenti letterari (temi, stile, simbolismi). Mappe geografiche interattive della Lombardia e dei percorsi dei personaggi. Esercizi di comprensione intermedi: Domande vero/falso sui fatti essenziali. Scelta multipla su dettagli di trama e personaggi. Riordino cronologico di eventi significativi (drag & drop). Quiz finale modulare: Domande a punteggio crescente su tre livelli di difficoltà (Facile, Intermedio, Difficile). Feedback immediato (“Corretto!”, “La risposta corretta è…”) con breve spiegazione. Meccaniche di gamification Progressione a livelli: Ogni modulo completato sblocca il successivo e “pagine” virtuali del manoscritto. Livelli tematici: “Introduzione”, “Conflitti”, “Risoluzioni”, “Epistole finali”. Sistema di punti (“Punti Alex”) Utenti guadagnano Punti Alex per risposte corrette e completamento giornaliero. Bonus giornalieri per login consecutivi (streak di accessi). Moltiplicatori di Punti Alex in caso di performance sopra soglia (es. ≥ 80% di esattezza nel quiz). Badge e ricompense digitali Badge tematici (es. “Bravo di Don Rodrigo”, “Novizio dei Cappuccini”, “Attore della Provvidenza”). Badge speciali per sfide settimanali e maratone di studio. Leaderboard e sfide social Classifica settimanale di Punti Edo a livello globale e tra amici. Possibilità di invitare amici a sfidarsi in quiz “head-to-head”. Profilo utente e personalizzazione Registrazione e autenticazione Email/password o social login (Google, Facebook). Verifica email obbligatoria per attivare l’account. Dashboard personale Sintesi dei progressi: livello attuale, Punti Alex totali, badge guadagnati. Statistiche personali: Accuracy media (%) Tempo medio di risposta per quiz Moduli con più errori (per suggerire ripetizioni). Sezione “Consigli personalizzati”: “Ripeti i capitoli con errori > 30%” “Prova quiz avanzato su personaggi secondari” “Rivisita la sezione storica di Leccco e Milano”. Notifiche e reminder Email e push notification (configurabili dall’utente) per: Promemoria giornaliero per riprendere l’apprendimento. Promemoria per sfida settimanale con amici. Messaggi motivazionali (“Hai collezionato 5 giorni consecutivi, continua così!”). Notifica di sblocco di nuovi moduli o badge. Sistema di supporto e feedback Chatbot di base (faq-driven) per chiarimenti sul testo: Spiegazioni brevi su termini arcaici o contesti storici. Link alle pagine di approfondimento nel modulo “Approfondisci”. Funzione “Segnala un errore” Modulo semplice dove l’utente può inviare osservazioni su contenuti errati o imprecisioni. Sistema di ticketing interno per il team contenuti.

5. Architettura e stack tecnologico (indicativi) Frontend Framework JavaScript: React (preferibile) o Vue.js, con attenzione all’accessibilità (standard WCAG AA). CSS Modulabile: Tailwind CSS per design pulito, responsive e tematizzabile. Gestione stato globale: Redux o Context API (se React). Componenti UI: libreria di componenti personalizzata (pulsanti, card, modali) con design coerente e responsive. Backend Tecnologia: Node.js con Express JS (o Nest.js per architettura più modulare) Oppure Python con Django o Flask + Flask-RESTful Architettura API: RESTful per tutte le operazioni CRUD su contenuti, progressi utente, autenticazione. Autenticazione: JWT (JSON Web Token) per sessioni stateless + refresh token. Database Relazionale: PostgreSQL per gestire dati utente, progressi, contenuti testuali, quiz, statistiche. Caching realtime: Redis o Memcached per: Salvataggio temporaneo delle sessioni Leaderboard in tempo reale (classifiche) Caching delle pagine statiche di modulo per velocizzare le richieste. Integrazione di servizi esterni Email transactionale: SendGrid o Mailgun per invio verification email, reminder, newsletter. Auth sociale: OAuth 2.0 con Google e Facebook. Notifiche push: Firebase Cloud Messaging (FCM) per web push e mobile push (in caso di futura app). Chatbot/FAQ: servizio bot (es. Dialogflow) per gestire risposte automatiche comuni; fallback a ticket interno. Infrastruttura e Hosting Cloud Provider: AWS, GCP o Azure (preferibile AWS). Containerizzazione: Docker + Kubernetes per orchestrazione (scalabilità e resilienza). Bilanciamento del carico: AWS ALB/ELB o equivalente. CDN: CloudFront o equivalente per distribuire asset statici (immagini, CSS, JS).

6. Flusso utente (User Flow) – esempi principali Registrazione/Login Utente atterra sulla landing page → clicca “Registrati” → inserisce email e password (oppure sceglie social login) → verifica email → accesso al primo onboarding. Onboarding iniziale Breve sondaggio: “Perché vuoi studiare I Promessi Sposi?” (es. ripasso scolastico, interesse personale, approfondimento). In base alla risposta, suggerimento dei moduli iniziali: Se “ripasso scolastico”: consiglia capitoli fondamentali (1–10). Se “approfondimento personale”: consiglia moduli con contesto storico e personaggi secondari. Impostazione obiettivo giornaliero (minimo 10 minuti) e preferenze notifiche. Accesso al primo modulo L’utente accede al Modulo 1 (Capitolo 1): Visualizza la lettura interattiva del testo, con evidenziazione di termini che mostrano glossario a comparsa. Sidebar “Approfondisci” attivabile per leggere contesto storico e mappa geografica. Alla fine della lettura, vengono proposti esercizi di comprensione (vero/falso, scelta multipla, riordino cronologico). Dopo gli esercizi, il quiz finale a tre livelli di difficoltà. Feedback istantaneo e assegnazione Punti Alex. Dashboard principale Mostra: Livello attuale (es. “Bravo di Don Rodrigo”, “Novizio dei Cappuccini”, “Attore della Provvidenza”). Prossimo traguardo (es. “Supera il 70% di accuracy nel modulo 2”). Sfida della settimana (es. “Rispondi correttamente a 5 domande sulla peste”). Amici online/invitati (indicazione di chi è disponibile per sfide “head-to-head”). Pulsanti: “Rivedi i capitoli già svolti” “Accedi al modulo successivo” (se sbloccato) “Visualizza statitiche personali” (grafico di progresso) Interazione sociale L’utente clicca “Invita un amico” → invia referral via link/email → l’amico si registra → entrambi ottengono bonus Punti Alex. Dashboard segnala: “Il tuo amico X ha accettato la sfida sul quiz del Capitolo 3 – partecipa ora!”. Possibilità di avviare quiz in tempo reale contro un amico con countdown.

7. Requisiti non funzionali Performance Tempo di caricamento iniziale < 3 secondi su rete mobile 3G per pagine principali (landing, modulo di lezione). Utilizzo di caching e CDN per ridurre latenza su contenuti statici. Query al database ottimizzate (indici, query parametrizzate). Scalabilità Architettura a microservizi (possibilità futura di separare chat, forum, analytics). Auto-scaling Kubernetes o container-based scaling per gestire picchi (es. periodo di esami a fine anno scolastico). Load balancing tra più istanze di backend. Sicurezza Protezione CSRF e XSS su tutte le form e endpoint. Crittografia dei dati sensibili (password hashing con bcrypt o Argon2). HTTPS obbligatorio su tutte le chiamate API. Politica di gestione token (access token breve, refresh token con revoca automatica). Conformità GDPR: Informativa privacy chiara, raccolta esplicita del consenso cookie. Possibilità di esportare o eliminare dati utente su richiesta (“diritto all’oblio”). Accessibilità Contrast ratio minimo WCAG AA per testi e sfondi. Testi alternativi (alt text) per tutte le immagini e icone. Navigazione via tastiera garantita (focus visibile, elementi interattivi accessibili). Lettore di schermo: markup semantico (heading, ARIA roles ove necessario).

8. Roadmap e milestone principali MVP (3 mesi) Landing page + sistema di registrazione utente (email/social). 3 moduli base di lettura interattiva (Capitoli 1–3). Quiz base e sistema Punti Edo semplice. Dashboard utente con monitoraggio progressi. Setup iniziale infrastruttura: database, API, hosting. Versione Beta (6 mesi) Pubblicazione di tutti i moduli (Capitoli 1–38). Implementazione delle meccaniche di gamification avanzata: badge, leaderboard globali. Notifiche via email/push configurabili dall’utente. Test di carico e ottimizzazioni performance (cache, query, CDN). Feedback da beta tester (studenti e insegnanti) e correzioni bug. Versione 1.0 (9 mesi) Funzioni social complete: Sfide “head-to-head” in tempo reale. Condivisione punteggi su social network (Facebook, Instagram). Chatbot di supporto integrato con base di conoscenza espansa. Integrazione multilingue per glossario (inglese, spagnolo) e interfaccia utente bilingue. Rilascio ufficiale su domini principali, avvio attività di marketing su social e scuole. Feature avanzate (12+ mesi) Forum di discussione interno: sezioni per capitolo, personaggi, contesto storico. Eventi in diretta con esperti di letteratura: webinar, Q&A su Zoom integrato. Moduli di approfondimento tematico: Analisi di contesti storici (peste, dominazioni straniere). Approfondimenti letterari (stile manzoniano, simbolismo religioso). Valutazione di un’app mobile nativa (iOS, Android) per notifiche native e offline reading.

9. Metriche di successo (KPI) DAU/MAU ratio (Daily Active Users / Monthly Active Users) ≥ 20%. Tasso di completamento modulo: ≥ 70% degli utenti completa ogni modulo entro 72 ore dall’apertura. Retention a 30 giorni: ≥ 20%. NPS (Net Promoter Score): ≥ +30 dopo 6 mesi di attività. Tasso di referral: ogni utente porta in media 0,5 nuovi utenti nei primi 3 mesi. Revenue mensile ricorrente (MRR): ≥ € 5.000 entro 12 mesi dal lancio.

10. Vincoli e rischi Vincoli Contenuti digitalizzati accurati: Necessità di validare la trascrizione integrale del testo con esperti di letteratura per evitare errori di digitazione. Budget limitato per produzione di contenuti: Creazione del glossario e del materiale di approfondimento richiede risorse editoriali dedicate. Tempo di sviluppo ristretto per MVP: Prioritizzare funzionalità essenziali (moduli base, quiz, dashboard). Rischi Precisione contenuti storici/letterari: Rischio di informazioni scorrette o incomplete; mitigazione con revisione continua da parte di esperti. Adozione iniziale inferiore alle aspettative: Possibile scarso coinvolgimento degli studenti; mitigare con attività di marketing mirate a scuole e docenti. Sostenibilità economica Rischio: modello freemium potrebbe non generare entrate sufficienti. Mitigazione: testare early pricing, offrire “punti extra” a pagamento e abbonamenti Premium con contenuti esclusivi (approfondimenti storici, webinar). Picchi di traffico in periodi critici (es. esami) Rischio di rallentamenti o downtime; mitigazione con infrastruttura auto-scalabile e test di carico regolari.

11. Consegne attese Documento PRD completo: in formato Word o Google Docs, strutturato secondo le sezioni sopra riportate. Diagramma dei flussi utente (User Flow): schema delle schermate principali e transizioni (registrazione, onboarding, modulo, quiz, dashboard). Wireframe di massima: bozza grafica per schermate chiave: Home/Landing page Pagina di registrazione/login Schermata modulo di lettura interattiva Schermata quiz (con feedback immediato) Dashboard utente con statistiche e leaderboard Piano di test e validazione: Definizione del processo di beta testing con gruppi di studenti e insegnanti Check-list per test funzionali (UX, flussi, integrazione API) e non funzionali (performance, sicurezza) Note aggiuntive per il team Il PRD è redatto in italiano, chiaro e conciso, con sezioni numerate per agevolare eventuali revisioni. Utilizzare elenchi puntati per facilitare la lettura; ricorrere a tabelle solo se strettamente necessario (ad esempio confronto funzionalità Free vs Premium). Mantenere tono professionale da Product Manager, motivando le scelte di design e indicando priorità e dipendenze quando rilevante. Durante lo sviluppo, fissare riunioni periodiche di sincronizzazione (sprint review ogni due settimane) per allineare il team sugli avanzamenti e sui rischi emergenti.