# Project Proposal: Development of "The Launch Pad" (TLP) Platform

Prepared by: TNET Development Team
Date: October 04, 2025
Version: 1.0

---

## Executive Summary

"The Launch Pad" (TLP), a comprehensive web platform and mobile app designed to serve as the ultimate global hub for rocket launches, space news, mission data, and orbital tracking. We are excited about this opportunity to build a scalable, user-centric solution that combines educational content, real-time visualizations, and interactive tools into a seamless experience.

Our proposal outlines a structured, phased approach to delivery, prioritizing the web platform as the MVP (Minimum Viable Product) before extending to the mobile app. This ensures rapid iteration, early value delivery, and alignment with your non-functional requirements (e.g., performance, security, accessibility, and SEO). We estimate a total development timeline of **~44 business days** and a budget of **$2650**, broken down into modular sprints for transparency and flexibility.

Key highlights:

- **Phased Delivery:** Five core modules for the web platform, followed by mobile app development and final integration.
- **Technical Approach:** Modern stack including React.js (frontend), Node.js/Express (backend), PostgreSQL (database), and integrations with mapping APIs (e.g., Leaflet/Google Maps) and 3D libraries (e.g., Three.js for Earth Navigator).
- **Value-Add Suggestions:** Where applicable, we'll propose UX optimizations, such as progressive web app (PWA) features for offline access on web and AI-driven personalization for launch recommendations.
- **Post-Launch Support:** Includes 3 days of free bug fixes and a handover package with documentation.

We look forward to your feedback and are available for a kickoff call to refine this proposal.

---

## Project Vision

TLP will empower space enthusiasts, professionals, and the general public with an intuitive platform that democratizes access to launch schedules, historical data, encyclopedic knowledge, and real-time orbital insights. By integrating dynamic visualizations (e.g., 3D

Earth models, interactive maps) with a robust admin backend, TLP will stand out as a reliable, engaging resource—optimized for high-traffic events like major launches.

The platform will launch as a responsive web application, with a React Native mobile app to follow, ensuring cross-device consistency. Future expansions (e.g., Live Operations) will be architected for easy integration without disrupting the core MVP.

## Project Scope Alignment

Our proposal fully aligns with your outlined scope:

- **Core Features:** Launch Center (calendars, stats, briefings), Space Database (TLPedia), Earth Navigator (3D viewer), Admin Platform.
- **Exclusions for MVP:** Live Operations module (designed as a plug-and-play extension).
- **Mobile App:** Mirrored web features with native enhancements (e.g., push notifications, offline mode).

We commit to suggesting UX improvements during discovery (e.g., gamified stats dashboards or AR previews for mobile) while respecting your provided designs.

## Technical Architecture Overview

To ensure scalability and maintainability:

- **Frontend:** React.js/vite; Tailwind CSS for styling; Three.js/Cesium for 3D visualizations.
- **Backend:** Node.js/Express API; PostgreSQL for relational data (launches, entities).
- **Data Model:** Normalized schema for entities (rockets, missions) with relationships; JSON fields for flexible timelines/media.
- **Integrations:** External APIs (e.g., NASA/The Space Devs for launch data, YouTube for streams); Mapping (Leaflet).
- **DevOps:** GitHub for version control; CI/CD via GitHub Actions.
- **Security:** JWT authentication; Role-based access control (RBAC); with best practices.
- **Mobile:** React Native for iOS/Android; Expo for rapid prototyping; Firebase for push notifications.

Full documentation

## Phased Delivery Plan: Timeline & Cost Estimate

We propose a modular sprint-based approach, delivering the web platform first (Modules 1-4) for early testing and feedback. Each module concludes with a demo, QA. The mobile app (Module 5) follows, with final integration (Module 6) ensuring end-to-end polish.

| Module | Description | Key Deliverables | Estimated Duration | Cost |
|---|---|---|---|---|
| **Module 1** | **Core Infrastructure & Launch Center Foundation** | - Backend setup (API, DB schema for launches).- Launch calendars (upcoming/previous) with filters.- Landing page and basic navigation.- Initial data ingestion scripts. | 5 Business Days | $300 |
| **Module 2** | **Launch Center Enhancements & Statistics** | - Mission briefings (overviews, timelines, maps).- Interactive stats dashboard (graphs via Chart.js).- Hazard zones and trajectory visualizations. | 5 Business Days | $400 |
| **Module 3** | **Space Database (TLPedia)** | - Category and profile pages (e.g., rockets, missions).- Search functionality and cross-links.- Media galleries and related entities. | 7 Business Days | $400 |
| **Module 4** | **Earth Navigator & Admin Platform** | - 3D orbital viewer with filters and object details.- Full admin dashboard (CRUD for data, workflows, media uploads).- RBAC, version history, and overrides. | 8 Business Days | $500 |
| **Module 5** | **Mobile App Development** | - React Native app mirroring web features.- Push notifications, offline access, user accounts.- iOS/Android builds with testing. | 15 Business Days | $850 |
| **Module 6** | **Integration, Testing & Launch** | - End-to-end testing (unit, integration, load).- SEO/Accessibility audits; PWA enhancements.- Documentation, and training. | 4 Business Days | $200 |
| **-** | **Future Expansions (e.g., Live Operations)** | - Modular hooks for ISS/Starbase integrations (quoted separately). | N/A | N/A |
| **Total** | | **Production-Ready Web + Mobile Platform** | **~44 Business Days** | **$2650** |

## Detailed Feature Breakdown

### Module 1: Core Infrastructure & Launch Center Foundation

- **Backend & Data Layer:** RESTful APIs for CRUD on launches; Data model with entities (e.g., Launch: id, date, provider, rocket, outcome).
- **Frontend Basics:** Responsive landing page with featured launches carousel; Calendar views (fullCalendar.js) filterable by date/provider/rocket/site/orbit.
- **UX Suggestions:** Infinite scroll for calendars to improve mobile performance.
- **Testing:** API unit tests; Basic accessibility checks.
- **Delivery:** This foundational phase focuses on creating the technical backbone of the platform. We will design a scalable data model for all entities (launches, rockets, missions, etc.), build the core API endpoints, and set up a secure admin interface. This ensures all subsequent modules are built on a solid and secure foundation.

### Module 2: Launch Center Enhancements & Statistics

- **Mission Briefings:** Per-launch pages with tabs for overview, timeline (sortable events), and maps (Leaflet for flat/hazard views; Google Earth Engine for trajectories).
- **Statistics Page:** Dynamic charts (launches/year, success rates, payload totals, reusability, orbits); Data aggregated via SQL queries with caching.
- **Links & Resources:** Embedded streams (YouTube API), press kits (file uploads).
- **UX Suggestions:** Hover tooltips on maps for stage events; Exportable CSV for stats.
- **Delivery:** We will bring the heart of the TLP experience to life. This includes building the dynamic launch calendars with robust filtering capabilities and creating the content-rich Mission Briefing pages. The interactive statistics dashboard will provide users with powerful data visualization tools to explore launch trends.

### Module 3: Space Database (TLPedia)

- **Structure:** Category hubs with faceted search (e.g., Elasticsearch integration for speed); Profile pages with tabs (overview, specs table, gallery, relations).
- **Content Management:** Basic CMS hooks (expanded in Module 4); Cross-links auto-generated from DB relations.
- **Media:** Image/video galleries with lazy loading; Alt text enforcement.
- **UX Suggestions:** Breadcrumb navigation for deep entity dives; "Related Launches" widget.
- **Delivery:** This module focuses on building the encyclopedic component. We will create a flexible template for entity profile pages that can accommodate different types of information (specs, galleries, related items) and implement a powerful search feature to help users easily navigate the vast database.

### Module 4: Earth Navigator & Admin Platform

- **3D Viewer:** Cesium.js-based globe with real-time TLE data (NORAD API); Filters (name/orbit/altitude); Clickable popups linking to TLPedia.

- **Admin Dashboard:** React Admin framework; Role-based views (e.g., writers see drafts); Workflow states (draft/review/publish); Media uploader with credits; Override toggles for API data.
- **Security Features:** Audit logs; Rollback via DB snapshots.
- **UX Suggestions:** Gesture controls for 3D (pinch-zoom); Dark mode for admin.
- **Delivery:** In this phase, we will develop the visually stunning Earth Navigator, fetching and displaying real-time orbital data. We will also integrate the 2D mapping solutions for mission briefings. On the backend, we will finalize the complete editorial workflow within the admin panel and implement technical SEO to ensure maximum visibility.

## Module 5: Mobile App Development

- **Core Screens:** Mirrored Launch Center, TLPedia search/browse, simplified 3D viewer (2D fallback for performance).
- **Native Features:** Expo Notifications for reminders; AsyncStorage for offline; Biometric login.
- **Testing:** Device-specific QA (iOS Simulator, Android Emulator); App Store prep.
- **UX Suggestions:** Swipe gestures for calendars; Haptic feedback on launch alerts.
- **Delivery:** Leveraging the APIs built for the web, we will develop the React Native mobile app. The focus will be on a clean, native-feeling user experience, implementing key features like push notifications and offline caching to provide value on the go.

## Module 6: Integration, Testing & Launch

- **Full Workflow:** E2E user journeys (e.g., search → briefing → 3D track).
- **Optimizations:** Lazy loading for 3D; CDN for media; i18n setup (React-i18next).
- **Handover:** API docs, admin guide, source code repo, 3-day support.
- **Delivery:** This final module is dedicated to quality assurance. We will conduct rigorous testing across all devices, optimize for speed, and perform security and accessibility audits.