Developer: Tiffany Morgan
Project Name: Launch Commander
Course: INEW 2334
Date: February 3, 2026
Project Type: Full-stack MEAN Application

Launch Commander - Communication Brief

Table of Contents

1. Project Summary

Launch Commander is a full-stack web application designed to transform rocket launch tracking into an interactive, competitive experience. Built with the MEAN stack (MongoDB, Express.js, Angular, Node.js), the application addresses the gap between passive launch observation and active audience engagement in the commercial spaceflight industry.

Core Problem:
Space enthusiasts currently lack platforms that combine real-time launch data with interactive prediction mechanics, resulting in passive consumption rather than engaged participation in the evolving space industry.

Solution:
Launch Commander gamifies launch tracking by allowing users to predict mission outcomes (success probability, MECO timing, orbit altitude, payload deployment) and compete on a scored leaderboard. The application integrates real-time data from the Launch Library 2 API to provide accurate upcoming launch schedules.

Technical Scope:
- Single-page application (SPA) architecture with Angular Material Design
- RESTful API backend for data management
- MongoDB database for predictions and user statistics
- Third-party API integration for live launch data
- Responsive design for desktop and mobile viewing

Primary Use Case:
Space enthusiasts visit the platform before upcoming launches, submit their predictions based on mission parameters, and return after launch completion to see their accuracy scores and leaderboard position.

Business Value

For Space Enthusiasts:
- Deeper engagement with space industry events
- Educational opportunity to learn mission parameters
- Community connection through competitive elements
- Recognition for aerospace knowledge via leaderboard rankings

For Developer Portfolio:
- Demonstrates full-stack development proficiency
- Showcases API integration skills
- Highlights data modeling and scoring algorithm design
- Proves UI/UX design capabilities with Angular Material
- Relevant to Firefly Aerospace internship application

Industry Relevance:
With 180+ orbital launches planned for 2026 (growth from 186 in 2023), the application taps into increasing public interest in commercial spaceflight driven by SpaceX, Blue Origin, and emerging providers like Firefly Aerospace.

2. Goals and Objectives

Short-Term Goals (Weeks 1-10: January - April 2026)

Goal 1: Deliver Functional MVP
- Objective: Complete all core features (launch display, prediction submission, scoring, leaderboard)
- Success Metric: User can make prediction and see score calculated correctly
- Timeline: End of Week 8 (April 11, 2026)

Goal 2: Ensure Production Readiness
- Objective: Deploy application to live hosting environment
- Success Metric: 99% uptime, <3s page load time
- Timeline: End of Week 9 (April 18, 2026)

Goal 3: Complete Academic Requirements
- Objective: Submit project with all required documentation for IMED 2345
- Success Metric: Pass capstone project with minimum 90% grade
- Timeline: Final submission April 30, 2026

Long-Term Goals (Post-Capstone: May - December 2026)

Goal 1: Expand User Base
- Objective: Reach 500 registered users within 6 months
- Strategy: Share on space subreddits, Twitter aerospace community, Discord servers
- Success Metric: 50 monthly active users by June 2026

Goal 2: Feature Enhancement
- Objective: Add user authentication and personalized profiles
- Features:

  - User registration/login system
  - Profile pages showing prediction history and statistics
  - Achievement badges (e.g., "Perfect Prediction", "SpaceX Expert")
  - Social features (friend leaderboards, prediction sharing)
- Timeline: July-September 2026

Goal 3: Monetization Exploration
- Objective: Assess revenue generation viability
- Options:
  - Premium tier with advanced analytics ($4.99/month)
  - Sponsored launch cards for aerospace companies
  - Educational partnerships with schools/planetariums
- Timeline: Q4 2026 (after user base establishment)

Goal 4: Industry Recognition
- Objective: Showcase project to potential employers (Firefly Aerospace, SpaceX, Blue Origin)
- Strategy: Include in internship applications, present at tech meetups
- Success Metric: Secure internship interview by August 2026

Goal 5: Launch Discord Community Server
- Objective: Build dedicated community space for Launch Commander users
- Features:
  - Launch announcement channels
  - Prediction discussion forums
  - Real-time launch watch parties
  - Bot integration for automated launch reminders
  - Leaderboard updates and achievement celebrations
  - User help/support channel
- Timeline: May-June 2026 (soft launch), expand Q3 2026
- Success Metric: 100+ Discord members by September 2026, 25% of registered users join server
- Moderation: Establish community guidelines, recruit volunteer moderators from top users

Measurable Success Indicator

Metric	Target (End of Capstone)	Long-Term Target (6 Months)
Functional Features Completed	100% (5/5)	100% + 3 new features
Page Load Time	<3 seconds	<2 seconds
Uptime	95%	99.5%
Registered Users	10 (test users)	500
Predictions Submitted	50+	2,000+
Mobile Responsiveness	100%	100%
Code Test Coverage	60%	80%

3. Digital Marketing Options

Search Engine Optimization (SEO)

On-Page SEO:

- Meta Tags:
  ```html
  <title>Launch Commander - Predict Rocket Launch Outcomes & Compete</title>
  <meta name="description" content="Track upcoming rocket launches, predict mission outcomes, and compete on the leaderboard. Real-time data from SpaceX, NASA, and more.">
  <meta name="keywords" content="rocket launches, SpaceX tracker, space predictions, launch schedule, aerospace game">
  ```
- Semantic HTML: Proper heading hierarchy (H1 → H2 → H3), alt text for images
- URL Structure: `launchcommander.app/launches`, `/leaderboard`, `/predictions`
- Schema Markup: JSON-LD structured data for events (launch dates)

Content Strategy:
- Blog Section (Future):
  - "How to Predict MECO Timing: A Beginner's Guide"
  - "Top 5 Most Challenging Launches to Predict in 2026"
  - "Understanding Orbital Altitude: LEO vs. GTO Missions"
- FAQ Page: Common questions about scoring, prediction mechanics, launch data sources
- Launch Previews: Detailed write-ups for major missions (Artemis, Starship, etc.)

Technical SEO:
- Sitemap.xml submission to Google Search Console
- robots.txt configuration
- HTTPS enforcement (SSL certificate)
- Mobile-friendly design (responsive CSS)
- Fast page load speed (<3s)

Expected Results:
- Rank on first page for "rocket launch predictions" within 3-6 months
- Organic traffic: 200-500 visitors/month by Q3 2026

Social Media Marketing

Platform 1: Twitter/X
- Target: Space enthusiasts, aerospace professionals, SpaceX fans
- Content Strategy:
  - Tweet countdown threads before major launches
  - Share leaderboard top performers weekly
  - Post launch prediction polls ("Will Falcon 9 B1234 land successfully?")
  - Use hashtags: SpaceX RocketLaunch SpaceNews Aerospace
- Posting Frequency: 3-5 times per week
- Example Tweet:
  > " Falcon 9 launching in 24 hours! Make your predictions on Launch Commander and compete for the top spot. Will it stick the landing? SpaceX LaunchPrediction [link]"

Platform 2: Reddit
- Target Subreddits:
  - r/spacex (1.2M members)
  - r/space (24M members)

  - r/aerospace (72K members)
  - r/webdev (1.8M members) - for technical showcase
- Content Strategy:
  - Share launch of application with demo video
  - Participate in launch discussion threads
  - Monthly "Top Predictors" posts
  - AMA (Ask Me Anything) about development process
- Rules Compliance: Check subreddit rules for self-promotion limits

Platform 3: YouTube (Long-Term)
- Content Ideas:
  - "How I Built Launch Commander: Full-Stack MEAN Tutorial"
  - "Top 10 Closest Predictions of 2026"
  - Live streams during major launches with community predictions
- Partnership Opportunity: Collaborate with space YouTubers (Everyday Astronaut, Scott Manley)

Platform 4: Discord
- Phase 1 (Immediate): Join existing space-focused servers
  - Servers: SpaceX Lounge, Space Enthusiasts, Everyday Astronaut Discord
  - Engagement: Share app in appropriate channels, participate in launch discussions
  - Strategy: Build reputation before launching own server
- Phase 2 (May-June 2026): Launch dedicated Launch Commander Discord server
  - Channels:
    - announcements (launch alerts, site updates)
    - launch-discussion (pre-launch analysis)
    - predictions (share strategies, ask for advice)
    - leaderboard-updates (weekly rankings, celebrations)
    - general-chat (space news, off-topic)
    - support (technical help)
    - live-events (watch parties during launches)
  - Bot Features:
    - Automated launch countdown timers
    - Prediction submission reminders
    - Leaderboard updates posted hourly during active launches
    - Achievement announcements (e.g., "@Tiffany just got a PERFECT PREDICTION! 🎯")
  - Community Events:
    - Monthly prediction contests with prizes (aerospace swag)
    - Launch watch parties with voice chat
    - AMAs with aerospace professionals (if connections made)
    - "Predictor of the Month" spotlights
  - Moderation: Community guidelines enforcing respectful discussion, volunteer moderator team from top users
  - Growth Strategy: Invite top 20 leaderboard users first (VIP early access), then open to all registered users
- Success Metrics:
  - 100+ members by September 2026
  - 25% of registered app users join Discord
  - Daily active users during major launch windows
  - User retention: 60% of members stay active after 30 days

Email Marketing

Email List Building:
- Newsletter signup form on homepage: "Get launch alerts & prediction reminders"
- Lead magnet: "Free Guide: Understanding Rocket Launch Success Factors"

Email Campaigns:
1. Launch Alert Emails:
   - Sent 48 hours before major launches
   - Subject: " SpaceX Starship Flight 8 launches tomorrow - Make your prediction!"
   - CTA: Link to prediction form

2. Weekly Digest:
   - Recap of launches from past week
   - Leaderboard highlights
   - Upcoming launches preview

3. Achievement Notifications:
   - "Congratulations! You ranked 5 this week"
   - Encourage sharing on social media

Tool: Mailchimp (free up to 500 subscribers) or SendGrid

Community Building

Strategies:
1. User-Generated Content: Encourage users to share predictions on social media with LaunchCommander hashtag
2. Monthly Contests: Prizes for top predictor (e.g., aerospace stickers, NASA merchandise)
3. Feature Requests: Public roadmap where users vote on new features
4. Ambassador Program: Recruit top users to promote in space communities

Paid Advertising (Future Consideration)

Google Ads:
- Budget: $200/month
- Target Keywords: "rocket launch schedule", "spacex tracker", "launch predictions"
- Expected CPC: $0.50-$1.50
- Estimated Clicks: 150-400/month

Reddit Ads:
- Budget: $100/month
- Target: r/spacex, r/aerospace subscribers
- Ad Format: Promoted post with demo GIF

ROI Timeline: Break-even on ads not expected until monetization phase (Q4 2026+)

4. Key Features

Feature 1: Real-Time Launch Tracking
Description: Display upcoming rocket launches from multiple providers in chronological order.

Functionality:
- Fetches data from Launch Library 2 API every 10 minutes
- Shows launch name, provider (SpaceX, NASA, etc.), vehicle type, launch site
- Displays countdown timer to launch time
- Color-coded status indicators (green = confirmed, yellow = pending, red = delayed)
- Mission images and descriptions

User Benefit: One-stop dashboard for all upcoming launches without visiting multiple provider websites.

Feature 2: Interactive Prediction System
Description: Users submit predictions for four mission parameters before launch.

Prediction Parameters:
1. Mission Success: Binary choice (successful orbit / failure)
2. MECO Timing: Estimated minutes to Main Engine Cutoff (numerical input)
3. Orbit Altitude: Predicted altitude in kilometers (numerical input)
4. Payload Deployment: Binary choice (successful / unsuccessful)

Constraints:
- One prediction per user per launch
- Predictions locked 10 minutes before launch time
- Form validation prevents duplicate submissions

User Benefit: Engages users beyond passive viewing, tests aerospace knowledge.

Feature 3: Intelligent Scoring Engine
Description: Calculates points based on prediction accuracy after launch completion.

Scoring Logic:
- Mission Success: 10 points (binary correct/incorrect)
- MECO Timing:
  - 5 points for within 30 seconds of actual time
  - 5 bonus points for perfect match (<5 seconds difference)
- Orbit Altitude:
  - 5 points for within 25km of actual altitude
  - 10 bonus points for perfect match (<10km difference)
- Payload Deployment: 10 points (binary correct/incorrect)
- Maximum Score: 40 points per prediction

User Benefit: Clear, objective performance measurement rewards aerospace expertise.

Feature 4: Competitive Leaderboard

Description: Ranked list of top users based on cumulative prediction scores.
Display Information:
- Username
- Total score (sum of all predictions)
- Number of predictions submitted
- Average accuracy rate

- Current rank

Sorting Options:
- By total score (default)
- By number of predictions
- By accuracy percentage

User Benefit: Social proof, recognition, motivation to improve predictions.

Feature 5: Responsive User Interface
Description: Clean, intuitive design using Angular Material components.

Design Elements:
- Three-column layout (launches, prediction form, leaderboard)
- Card-based UI for visual separation
- Material Design color palette (blues, whites, grays)
- Smooth animations for form submissions
- Mobile-responsive (collapses to single column on small screens)
- Accessibility compliant (WCAG 2.1 AA standards)

User Benefit: Professional appearance, easy navigation, works on any device.

5. Target Audience

Primary Audience: Space Enthusiasts (Ages 18-45)

Demographics:
- Age: 18-45 years old (tech-savvy, active online)
- Gender: 70% male, 30% female (based on r/spacex demographics)
- Location: United States (60%), Europe (25%), Other (15%)
- Education: College students, graduates, professionals in STEM fields
- Income: $40K-$90K (disposable income for hobbies)

Psychographics:
- Interests: Space exploration, technology, science, engineering
- Behaviors:
  - Watches rocket launches live on YouTube
  - Follows SpaceX, NASA, Blue Origin on social media
  - Reads space news (Space.com, NASASpaceflight.com)
  - Plays simulation games (Kerbal Space Program)
- Pain Points:
  - Passive consumption of space content (wants interactive engagement)
  - Difficulty tracking launches across multiple providers
  - No outlet to test/showcase aerospace knowledge
- Motivations:
  - Learn more about spaceflight mechanics
  - Connect with like-minded community
  - Achieve recognition for expertise

User Personas:

Persona 1: "Alex the Engineering Student"
- 22 years old, Aerospace Engineering major at Texas A&M
- Watches every SpaceX launch, dreams of working at SpaceX or NASA
- Wants to apply classroom knowledge to real-world predictions
- Active on Reddit r/spacex, Twitter space community
- Uses app to test understanding of orbital mechanics

Persona 2: "Jordan the Space Fan"
- 34 years old, software developer, space enthusiast hobby
- Casually follows major launches but not deep technical knowledge
- Enjoys competitive leaderboards (plays fantasy sports)
- Uses app for fun competition with friends, learns about mission parameters

Secondary Audience: Aerospace Professionals

Demographics:
- Age: 25-55 years old
- Occupation: Engineers, scientists, technicians at aerospace companies
- Education: Bachelor's/Master's in engineering, physics, or related fields

Use Case:
- Light entertainment during workday
- Friendly competition with coworkers
- Stay updated on industry launch schedules

Tertiary Audience: Educators & Students

Demographics:
- Teachers at middle/high schools teaching STEM subjects
- Students in aerospace/engineering programs

Use Case:
- Educational tool to teach mission parameters
- Classroom activity for upcoming launches
- Demonstrates real-world application of physics concepts

Audience Growth Strategy

Phase 1 (Months 1-3): Focus on early adopters in space subreddits and Twitter
Phase 2 (Months 4-6): Expand to YouTube space channels, Discord communities
Phase 3 (Months 7-12): Partner with educators for classroom use, aerospace company partnerships

6. Competitors Analysis

Competitor 1: Next Spaceflight
Website: https://nextspaceflight.com
Overview:
Comprehensive launch schedule tracker showing upcoming and past launches from all global providers. Includes countdowns, mission details, and livestream links.

Strengths:
- Extensive database (tracks 200+ launches per year)
- Clean, professional interface
- Mobile apps (iOS/Android)
- Notifications for upcoming launches
- Historical launch archive dating back decades
- Large established user base (500K+ monthly visitors)

Weaknesses:
- Purely informational (no interactive features)
- No user engagement mechanics (no predictions, no gamification)
- Limited social features (no community aspect)
- Revenue model unclear (minimal monetization)

Competitive Advantage for Launch Commander:
- Interactivity: Prediction system creates active participation vs. passive viewing
- Gamification: Leaderboard and scoring incentivize return visits
- Community: Competitive elements foster user engagement

Competitor 2: Space Launch Now
Website: https://spacelaunchnow.me

Overview:
Launch tracking application (web + mobile apps) with detailed mission information, vehicle specs, and agency profiles. Open-source project.

Strengths:
- Excellent mobile apps (4.5+ stars on app stores)
- Push notifications for launches
- Vehicle database with 3D models
- Astronaut profiles
- Launch site maps
- Free and ad-free
- API access for developers

Weaknesses:
- No gamification or user engagement features
- No prediction or scoring system
- Limited monetization (donation-based)
- Crowded UI with excessive information

Competitive Advantage for Launch Commander:
- Focus: Streamlined experience centered on predictions, not information overload
- Scoring System: Objective measurement of user knowledge
- Web-First: Accessible without app download (lower barrier to entry)

Competitor 3: SpaceX Website Launch Schedule
Website: https://www.spacex.com/launches/

Overview:

Official SpaceX launch schedule page showing upcoming Falcon 9, Falcon Heavy, and Starship missions.

Strengths:
- Authoritative source (directly from SpaceX)
- High-quality mission imagery and videos
- Detailed mission descriptions
- Livestream integration

Weaknesses:
- Limited Scope: Only SpaceX launches (no NASA, ULA, Rocket Lab, etc.)
- No interactivity whatsoever
- No community features
- Primarily marketing tool, not engagement platform

Competitive Advantage for Launch Commander:
- Multi-Provider: Tracks all launch providers, not just SpaceX
- Interactivity: Prediction and competition features
- Community: Leaderboard creates social engagement

Competitor 4: Rocket Launch Live
Website: https://www.rocketlaunch.live

Overview:
Launch tracking site with live countdowns, mission details, and launch provider information.

Strengths:
- Real-time countdown timers
- Embedded livestreams
- Clean, modern design
- Social media integration

Weaknesses:
- Limited interactivity (some polls, but not systematic predictions)
- No scoring system or leaderboard
- Monetization through ads (impacts user experience)

Competitive Advantage for Launch Commander:
- Structured Predictions: Four specific parameters vs. simple polls
- Persistent Scoring: Long-term leaderboard vs. one-off polls
- Ad-Free Experience: Cleaner interface (initially)

Competitor 5: Everyday Astronaut Community
Platform: YouTube + Discord

Overview:
Tim Dodd (Everyday Astronaut) has built large space enthusiast community through YouTube videos and Discord server (50K+ members).

Strengths:

- Massive engaged audience (1.4M YouTube subscribers)
- Live launch coverage with expert commentary
- Active Discord community discussions
- Educational content explaining rocket science

Weaknesses:
- Not a dedicated application (fragmented across platforms)
- No formal prediction or scoring system
- Discord predictions are informal, not tracked
- Video content, not interactive tool

Competitive Advantage for Launch Commander:
- Dedicated Platform: Purpose-built for predictions
- Automated Scoring: No manual tracking needed
- Persistent Statistics: Long-term user profile and history
- Potential Partnership: Could collaborate with Everyday Astronaut for promotion

Market Positioning

Competitive Matrix:

Feature	Launch Commander	Next Spaceflight	Space Launch Now	SpaceX Site	Rocket Launch Live
Multi-Provider Tracking	O	O	O	x	O
Prediction System	O	x	x	x	Limited (polls)
Scoring Algorithm	O	x	x	x	x
Leaderboard	O	x	x	x	x
Mobile App	x (future)	O	O	x	x
Historical Archive	x (future)	O	O	O	Limited
Push Notifications	x (future)	O	O	x	x

Unique Value Proposition:
"Launch Commander is the only platform that transforms rocket launch tracking into an interactive, competitive experience with structured predictions and performance scoring."

Competitive Strategy

Short-Term (2026):
- Focus on prediction mechanics (competitors lack this entirely)
- Build community around leaderboard competition
- Target Reddit/Discord communities where existing competitors have weak presence

Long-Term (2027+):
- Add mobile apps to compete with Space Launch Now
- Expand historical data to compete with Next Spaceflight
- Develop API for third-party integrations
- Premium features (advanced analytics) for monetization

Defensibility:
- Proprietary scoring algorithm
- User-generated prediction data creates network effects
- Community engagement hard for competitors to replicate

7. Branding Options

Brand Identity

Brand Name: Launch Commander

Tagline Options:
1. "Predict. Compete. Conquer the Cosmos." (primary)
2. "Your Mission: Predict Every Launch"
3. "Track. Predict. Lead the Leaderboard."

Brand Personality:
- Innovative: Cutting-edge technology, forward-thinking
- Competitive: Encourages friendly rivalry, achievement-focused
- Accessible: Welcoming to beginners and experts alike
- Authoritative: Reliable data, professional presentation
- Exciting: Captures thrill of space exploration

Visual Identity

Color Palette

Primary Colors:
- Deep Space Blue: `0A1828` (backgrounds, headers)
  - Evokes night sky, space, professionalism
- Rocket Red: `BF211E` (CTAs, active states, accents)
  - High-energy, attention-grabbing, urgent
- Stellar White: `F5F5F5` (text, cards, content backgrounds)
  - Clean, modern, high contrast

Secondary Colors:
- Orbit Orange: `FF6B35` (highlights, success states)
  - Friendly, warm, celebratory (used for achievements)
- Nebula Purple: `4A4E69` (secondary elements, borders)
  - Depth, sophistication, differentiation
- Launch Pad Gray: `9A8C98` (disabled states, dividers)
  - Subtle, neutral, supports hierarchy

Color Usage Guidelines:
- Backgrounds: Deep Space Blue (0A1828)
- Cards/Content: Stellar White (F5F5F5)
- Primary Buttons: Rocket Red (BF211E)
- Links: Orbit Orange (FF6B35)
- Text: Deep Space Blue on white, Stellar White on dark

Accessibility:
- All color combinations meet WCAG 2.1 AA contrast ratio (4.5:1 minimum)
- Never rely solely on color to convey information (use icons/text labels)

Typography

Heading Font: Rajdhani (Google Fonts)
- Geometric sans-serif with futuristic feel
- Weights: Bold (700) for H1/H2, SemiBold (600) for H3/H4
- Usage: Page titles, section headers, leaderboard names
- Example: "LAUNCH COMMANDER" in 700 weight

Body Font: Inter (Google Fonts)
- Clean, highly readable sans-serif
- Weights: Regular (400) for body text, Medium (500) for emphasis
- Usage: Paragraphs, form labels, card content
- Excellent readability at small sizes (mobile-friendly)

Monospace Font: Roboto Mono (Google Fonts)
- For technical data (countdown timers, coordinates, scores)
- Weight: Regular (400)
- Usage: Timers, numerical data, code snippets

Font Scale:
- H1: 36px (Rajdhani Bold)
- H2: 28px (Rajdhani SemiBold)
- H3: 22px (Rajdhani SemiBold)
- Body: 16px (Inter Regular)
- Small: 14px (Inter Regular)
- Captions: 12px (Inter Medium)

Logo Concepts

Concept 1: "Trajectory Icon"
- Upward-curving arrow with rocket silhouette at tip
- Geometric, minimalist design
- Works in single color (white on dark, dark on light)
- Symbolizes: Launch trajectory, upward progress, prediction path

Concept 2: "Commander Badge"
- Circular badge with star in center
- Rocket silhouette crossing horizontally
- Military-inspired authority aesthetic
- Symbolizes: Achievement, rank, leadership

Concept 3: "LC Monogram"
- Stylized "LC" letters integrated with launch pad structure
- Modern, tech-forward design
- Scalable to small sizes (favicon)
- Symbolizes: Brand name, infrastructure, foundation

Recommended: Concept 1 (Trajectory Icon) for versatility and immediate recognition

Logo Variations:
- Full logo: Icon + "Launch Commander" wordmark
- Icon-only: For favicon, social media profile pictures
- Wordmark-only: For horizontal headers, emails

UI Design Patterns

Card Design:
- Rounded corners (8px border-radius)
- Subtle drop shadow (0px 2px 8px rgba(0,0,0,0.1))
- White background on dark page
- 16px internal padding

Buttons:
- Primary (CTA): Rocket Red background, white text, 4px border-radius, hover darkens
- Secondary: Outlined (2px border), transparent background, hover fills
- Disabled: Gray background, reduced opacity (0.5)

Forms:
- Material Design inputs (underline style)
- Floating labels
- Inline validation icons (checkmark/error)
- Error text in Rocket Red

Icons:
- Material Icons library (consistent style)
- 24px default size
- Paired with text labels (accessibility)

Spacing System:
- Base unit: 8px
- Small: 8px, Medium: 16px, Large: 24px, XL: 32px
- Consistent rhythm throughout UI

Brand Voice and Messaging

Tone:
- Enthusiastic but not juvenile: "Track upcoming launches and test your aerospace knowledge!"
- Technical but accessible: Explain MECO without overwhelming newcomers
- Competitive but friendly: "Can you top the leaderboard?" not "Crush your enemies!"

Sample Messaging:

Homepage Hero:
> Track Every Launch. Predict Every Outcome.
> Join the space community competing to make the most accurate rocket launch predictions. Real-time data. Real competition.
> [Get Started Free →]

Feature Section:
> Your Mission Control Dashboard
> See upcoming launches from SpaceX, NASA, ULA, and more. Make predictions on mission success, MECO timing, orbit altitude, and payload deployment. Earn points for accuracy and climb the leaderboard.

Call-to-Action:

> Ready to prove your aerospace expertise?
> [Make Your First Prediction →]

Brand Assets Checklist

Required Deliverables:
- [ ] Logo (SVG + PNG: color, white, black versions)
- [ ] Favicon (16x16, 32x32, 180x180 for iOS)
- [ ] Social media graphics (Twitter header: 1500x500, profile: 400x400)
- [ ] Open Graph image for link sharing (1200x630px)
- [ ] Email header template
- [ ] Style guide PDF (colors, fonts, logo usage rules)

8. Project Timeline with Deliverables

Week 1: Foundation & Planning (January 27-31, 2026)

Phase: Requirements Gathering & Setup

Deliverables:
- [x] Project Proposal Document (completed January 28)
- [x] Client Survey (40 questions, completed January 29)
- [x] Technical Documentation (completed January 29)
- [x] Communication Brief (completed January 29)
- [x] GitHub repository created with README
- [x] Project board with 6 initial tasks
- [x] Development environment confirmed (Node.js, MongoDB, Angular CLI)

Hours: 16 hours (documentation-heavy week)

Milestone: All Part 1 and Part 2 documentation complete

Week 2: Backend Development (February 3-7, 2026)

Phase: Server & Database Implementation

Deliverables:
- [ ] Express.js server with CORS configuration
- [ ] MongoDB schema design (launches, predictions, leaderboard collections)
- [ ] API endpoints:
  - `GET /api/launches` - Fetch upcoming launches from Launch Library 2 API
  - `POST /api/predictions` - Save user prediction with validation
  - `GET /api/predictions/:username` - Retrieve user's prediction history
  - `GET /api/leaderboard` - Return top 10 users by score
- [ ] Data validation middleware (prevent duplicate predictions)
- [ ] Error handling for API failures
- [ ] API testing with Postman (documented test cases)

Hours: 24 hours

Milestone: Backend fully functional, tested endpoints return correct data

Week 3: Frontend Structure (February 10-14, 2026)

Phase: Angular Application Setup

Deliverables:
- [ ] Angular project initialization with Material Design
- [ ] Component structure:
  - AppComponent (main layout)
  - LaunchListComponent (left column)
  - PredictionFormComponent (center column)
  - LeaderboardComponent (right column)
- [ ] Service layer for HTTP requests (ApiService)
- [ ] Data models (TypeScript interfaces for Launch, Prediction, LeaderboardEntry)
- [ ] Routing configuration (if multi-page future)
- [ ] Responsive grid layout (three columns collapse on mobile)

Hours: 20 hours

Milestone: Frontend displays sample data, layout responsive on mobile

Week 4: Feature Integration (February 17-21, 2026)

Phase: Connect Frontend to Backend

Deliverables:
- [ ] API service integration (HTTP requests to backend)
- [ ] Launch cards displaying real API data
- [ ] Prediction form with validation:
  - Username required
  - MECO timing: 5-20 minutes range
  - Orbit altitude: 100-40000 km range
  - Radio buttons for binary choices
- [ ] Form submission handling (success/error messages)
- [ ] Leaderboard populating from database
- [ ] Countdown timers showing time until launch
- [ ] Loading states (spinners while fetching data)

Hours: 22 hours

Milestone: Full user flow functional (view launch → submit prediction → see leaderboard)

Week 5: Scoring & Refinement (February 24-28, 2026)

Phase: Scoring Logic & Polish

Deliverables:
- [ ] Scoring algorithm implementation (backend)
  - Mission success: 10 points
  - MECO timing: 5 points + 5 bonus (within 30s / perfect)
  - Orbit altitude: 5 points + 10 bonus (within 25km / perfect)
  - Payload deployment: 10 points

- [ ] Manual scoring script for testing (updateLaunch.js)
- [ ] User prediction history display showing points breakdown
- [ ] Leaderboard sorting and ranking logic
- [ ] UI polish:
  - Consistent spacing and colors
  - Hover states on interactive elements
  - Success animations on form submission
  - Error handling UI (toast notifications)
- [ ] Cross-browser testing (Chrome, Firefox, Safari, Edge)

Hours: 24 hours

Milestone: Scoring calculations correct, UI polished and professional

Week 6: User Authentication (March 3-7, 2026)

Phase: Security Implementation

Deliverables:

- [ ] Backend authentication setup:
  - Install bcrypt and jsonwebtoken packages
  - POST /api/register endpoint (user registration with password hashing)
  - POST /api/login endpoint (JWT token generation)
  - GET /api/profile endpoint (protected route)
  - Authentication middleware for JWT verification
  - Update predictions endpoint to require authentication
- [ ] Frontend authentication UI:
  - Login form component
  - Registration form component
  - Auth service for token management
  - Store JWT in localStorage
  - Update prediction submission to send auth token
- [ ] MongoDB users collection schema
- [ ] Password validation (minimum 6 characters)
- [ ] Security testing (prevent unauthorized predictions)

Hours: 20 hours

Milestone: Users must login to submit predictions, passwords securely hashed

Week 7: Advanced Features & Polish (March 10-14, 2026)

Phase: Feature Enhancement

Deliverables:

- [ ] User profile page showing personal statistics
- [ ] Enhanced leaderboard with user badges/achievements
- [ ] Prediction history with detailed score breakdowns
- [ ] Admin dashboard for manual launch updates
- [ ] Email validation on registration
- [ ] "Forgot password" placeholder (future implementation)
- [ ] Loading states and error handling improvements

- [ ] Accessibility improvements (ARIA labels, keyboard navigation)
- [ ] Mobile UX refinements

Hours: 22 hours

Milestone: Application feature-complete with polished UX

Week 8: Testing & Bug Fixes (March 17-21, 2026)

Phase: Quality Assurance

Deliverables:
- [ ] Comprehensive testing:
  - User authentication flow (register, login, logout)
  - Prediction submission with various data inputs
  - Scoring algorithm verification with real launch data
  - Leaderboard accuracy testing
  - Edge case handling (expired tokens, invalid inputs)
- [ ] Cross-browser compatibility testing (Chrome, Firefox, Safari, Edge)
- [ ] Mobile device testing (iOS Safari, Android Chrome)
- [ ] Performance optimization:
  - Database query optimization
  - Frontend bundle size reduction
  - Image compression
  - Lazy loading implementation
- [ ] Bug fixes and refinements
- [ ] Code cleanup and commenting

Hours: 24 hours

Milestone: All critical bugs resolved, performance optimized

Week 9: Deployment & Documentation (March 24-28, 2026)

Phase: Production Deployment

Deliverables:
- [ ] Backend deployed to Railway/Render
- [ ] Frontend deployed to Vercel/Netlify
- [ ] MongoDB Atlas production database setup
- [ ] Environment variables configured
- [ ] Domain name registered and DNS configured
- [ ] SSL certificate active (HTTPS)
- [ ] Production testing:
  - End-to-end user flow
  - Mobile responsiveness verification
  - Performance testing (Lighthouse score >90)
- [ ] GitHub repository finalized:
  - Clean commit history
  - README with setup instructions
  - Instructor added as collaborator

- [ ] Project board updated (mark completed tasks)
- [ ] Initial documentation review
- [ ] Production smoke testing

Hours: 24 hours

Milestone: Live production application deployed and stable

Week 10: Final Polish & Submission (March 31 - April 4, 2026)

Phase: Final Review & Capstone Submission

Deliverables:
- [ ] Final code review and refactoring
- [ ] Complete documentation update:
  - README with full setup instructions
  - API documentation
  - Architecture diagrams
  - User guide
- [ ] Demo video recording (3-5 minutes showing key features)
- [ ] Presentation slides preparation (if required)
- [ ] Final testing round:
  - Complete user journey testing
  - Performance verification (Lighthouse score >90)
  - Security audit checklist
- [ ] Code repository cleanup:
  - Remove debug code and console logs
  - Ensure .env file excluded from Git
  - Clean commit history
- [ ] Project portfolio page creation
- [ ] Final submission package:
  - All documentation in Google Docs format
  - GitHub repository link
  - Live application URL
  - Demo video
- [ ] Submit to Canvas before deadline

Hours: 20 hours

Milestone: Complete project submitted, portfolio-ready

Total Timeline: 10 Weeks (212 Hours)
Start Date: January 27, 2026
End Date: April 4, 2026
Final Submission Deadline: April 27, 2026, 11:59 PM

Gantt Chart Representation

| Week | Dates | Phase | Key Activities | Hours |

1	Jan 27-31	Foundation	Documentation, GitHub setup	16
2	Feb 3-7	Backend	Express API, MongoDB schema, endpoints	24
3	Feb 10-14	Frontend	Angular components, services, layout	20
4	Feb 17-21	Integration	Connect frontend/backend, forms, API calls	22
5	Feb 24-28	Scoring	Algorithm implementation, UI polish, testing	24
6	Mar 3-7	Authentication	JWT, bcrypt, login/register endpoints	20
7	Mar 10-14	Features	User profiles, badges, admin dashboard	22
8	Mar 17-21	Testing	QA, bug fixes, performance optimization	24
9	Mar 24-28	Deployment	Production hosting, DNS, SSL setup	24
10	Mar 31-Apr 24	Submission	Final docs, demo video, Canvas submission	20

Risk Mitigation

Risk 1: API Data Unavailability
- Mitigation: Cache launch data locally, use mock data if API down
- Buffer: 2 hours built into Week 7 for contingencies

Risk 2: Hosting Issues
- Mitigation: Test deployment early in Week 8, have backup hosting provider ready
- Buffer: 4 hours allocated for troubleshooting

Risk 3: Scope Creep
- Mitigation: Strictly adhere to MVP features, track "nice-to-have" separately
- Decision: No new features after Week 4 unless ahead of schedule

9. Project Budget

Development Costs

Labor:
- Developer Time: 132 hours @ $30/hour (student rate)
- Total Development Cost: $3,960

Cost Breakdown by Phase:

Phase	Hours	Cost @ $30/hr
Planning & Documentation	16	$480
Backend Development	24	$720
Frontend Development	20	$600
Integration	22	$660
Scoring & Testing	24	$720
Deployment & Documentation	26	$780
Total	132	$3,960

(Note: This is theoretical cost for portfolio/academic project; no actual client billing)

Infrastructure Costs

Year 1 (2026):

Service	Provider	Cost	Purpose
Domain Registration	Namecheap	$12/year	launchcommander.app domain

Frontend Hosting	Vercel	$0/month (free tier)	Angular SPA hosting
Backend Hosting	Railway	$5/month ($60/year)	Node.js API server
Database	MongoDB Atlas	$0/month (free 512MB)	Launch/prediction data storage
SSL Certificate	Automatic (Vercel/Railway)	$0	HTTPS security
Error Tracking	Sentry	$0/month (free 5K events)	Error monitoring
Email Service	SendGrid	$0/month (free 100 emails/day)	Launch notifications (future)
Total Year 1		$72/year	

Monthly Recurring: $5/month ($60/year) + $12 domain = $72 total

Year 2+ (Scaling Scenario):

If application grows to 5,000+ users:

Service	Provider	Cost	Purpose
Domain	Namecheap	$12/year	Domain renewal
Frontend Hosting	Vercel Pro	$20/month ($240/year)	Increased bandwidth, SLA
Backend Hosting	Railway Pro	$20/month ($240/year)	More compute, guaranteed uptime
Database	MongoDB Atlas M10	$57/month ($684/year)	Dedicated cluster, backups
Error Tracking	Sentry Team	$26/month ($312/year)	More events, team features
Email Service	SendGrid Essentials	$20/month ($240/year)	40K emails/month
Total Year 2		$1,728/year	

Monthly Recurring: $144/month

Total Project Investment (Academic Context)

Capstone Project (6 Weeks):
- Development Time: $3,960 (theoretical)
- Infrastructure: $6 (one month hosting during development)
- Total: $3,966

First Year Operation (Post-Capstone):
- Ongoing hosting: $72/year
- Maintenance time: ~5 hours/month @ $30/hour = $1,800/year
- Total Year 1: $1,872

Break-Even Analysis (If Monetizing):
- Premium subscription: $4.99/month
- Need 30 paying users to cover $144/month hosting (scaled infrastructure)
- Need 376 paying users to cover $30/hour maintenance time (5 hrs/mo)

Recommendation: Keep free for first year (portfolio/resume value exceeds revenue potential)

10. Time Estimation

Detailed Task Breakdown

Phase 1: Planning & Documentation (16 hours)

Task	Estimated Hours	Actual Hours	Notes

Project proposal writing	4	4	Executive summary, objectives, technical approach
Client survey creation (40 questions)	3	3	8 sections, research question types
Technical documentation	4	4	Hosting, database, analytics sections
Communication brief	3	3	Competitors, branding, marketing strategy
GitHub repository setup	1	-	Create repo, README, .gitignore
Project board configuration	1	-	Create 6 tasks with descriptions
Subtotal	16	14	2 hours buffer remaining

Phase 2: Backend Development (24 hours)

Task	Estimated Hours	Actual Hours	Notes
Express server setup & CORS config	2	2	Initialize npm, install dependencies
MongoDB schema design	2	2	Design 3 collections with indexes
Launch Library 2 API integration	3	4	Research API, handle pagination, caching
GET /api/launches endpoint	2	2	Fetch and format launch data
POST /api/predictions endpoint	4	5	Validation, duplicate prevention, save to DB
GET /api/predictions/:username endpoint	2	1.5	Query by username, return sorted results
GET /api/leaderboard endpoint	3	2	Aggregation pipeline, calculate rankings
Error handling middleware	2	1	Try-catch blocks, error responses
API testing (Postman)	4	3	Create test collection, document endpoints
Subtotal	24	22.5	1.5 hours ahead of schedule

Phase 3: Frontend Development (20 hours)

Task	Estimated Hours	Actual Hours	Notes
Angular CLI initialization	1	1.5	Create project, install Angular Material
Component creation (4 components)	3	3	App, LaunchList, PredictionForm, Leaderboard
API service layer	2	2	HTTP methods for each endpoint
TypeScript interfaces/models	1	1	Launch, Prediction, LeaderboardEntry types
Responsive grid layout (CSS)	4	5	Three-column desktop, single-column mobile
Material Design theming	2	2	Custom color palette, typography
Routing configuration	1	0.5	Basic setup for future multi-page
Testing with mock data	3	3	Ensure components render correctly
Accessibility review (ARIA labels)	2	2	Screen reader testing, keyboard navigation
Cross-browser testing	1	1	Chrome, Firefox, Safari, Edge
Subtotal	20	21	1 hour over estimate

Phase 4: Integration (22 hours)

Task	Estimated Hours	Actual Hours	Notes
Connect LaunchList to backend API	3	2.5	HTTP GET, display in cards
Prediction form validation logic	4	4	Required fields, range checks, error messages
Form submission HTTP POST	3	3.5	Send data, handle success/error responses

Leaderboard data fetching	2	1.5	GET request, populate table
Countdown timer implementation	3	4	Calculate time difference, update every second
Loading states (spinners)	2	2	Show while fetching data
Success/error notifications (toasts)	2	2	Material Snackbar component
User prediction history display	3	3	Fetch and render past predictions
Subtotal	22	23	1 hour over estimate

Phase 5: Scoring & Refinement (24 hours)

Task	Estimated Hours	Actual Hours	Notes
Scoring algorithm (backend logic)	5	-	Calculate points for each parameter
updateLaunch.js script	3	-	Manual tool to input actual results
Test scoring with sample data	2	-	Verify calculations correct
Points breakdown display (frontend)	3	-	Show how score was calculated
Leaderboard sorting/ranking	2	-	Ensure correct order, handle ties
UI polish (spacing, colors, shadows)	4	-	Consistent design system
Hover states and animations	2	-	Smooth transitions on interactions
Mobile responsiveness fixes	2	-	Test on iPhone, Android, tablet
Performance optimization	1	-	Lazy loading, minimize bundle size
Subtotal	24	0	Not started

Phase 6: Deployment & Documentation (26 hours)

Task	Estimated Hours	Actual Hours	Notes
MongoDB Atlas setup	2	-	Create cluster, configure access
Backend deployment (Railway)	3	-	Connect repo, set env vars, deploy
Frontend deployment (Vercel)	2	-	Connect repo, configure build settings
Domain registration & DNS	2	-	Purchase domain, point to hosting
SSL certificate verification	1	-	Ensure HTTPS working
Production testing (end-to-end)	4	-	Full user flow on live site
Lighthouse performance audit	1	-	Target score >90
GitHub repo finalization	2	-	Clean commits, update README
Add instructor as collaborator	0.5	-	Send invitation
Project board updates	1	-	Mark completed tasks, add notes
Demo video recording	3	-	Screen recording with narration
Final documentation review	2	-	Proofread all documents
Canvas submission	0.5	-	Upload files, submit assignment
Buffer for unexpected issues	2	-	Troubleshooting reserve
Subtotal	26	0	Not started

Total Time Investment

Planned: 132 hours

Breakdown by Category:

- Backend: 22.5 hours (complete)
- Frontend: 21 hours (complete)
- Integration: 23 hours (complete)
- Documentation: 14 hours (complete)
- Scoring: 24 hours (remaining)
- Deployment: 26 hours (remaining)

Weekly Time Commitment:
- Weeks 1-4: ~22 hours/week (completed)
- Weeks 5-6: ~25 hours/week (remaining)
- Average: 22 hours/week (equivalent to part-time job)

Time Management Strategy

Daily Schedule (Example Week 5):
- Monday: 4 hours (scoring algorithm backend)
- Tuesday: 4 hours (continue scoring logic)
- Wednesday: 5 hours (updateLaunch.js script + testing)
- Thursday: 4 hours (frontend points breakdown display)
- Friday: 4 hours (UI polish, spacing, colors)
- Saturday: 3 hours (animations, hover states)
- Sunday: Buffer / catch-up

Productivity Tools:
- Pomodoro Technique (25-minute focused sessions)
- Task tracking in GitHub Project board
- Daily commit goal (minimum 1 meaningful commit)
- Weekly progress review (compare actual vs. estimated)

Time Estimation Accuracy Review

Backend Phase:
- Estimated: 24 hours
- Actual: 22.5 hours
- Variance: -6% (came in under budget due to efficient API integration)

Frontend Phase:
- Estimated: 20 hours
- Actual: 21 hours
- Variance: +5% (layout took longer than expected due to CSS Grid learning curve)

Integration Phase:
- Estimated: 22 hours
- Actual: 23 hours
- Variance: +4.5% (countdown timer more complex than anticipated)

Overall Accuracy: ±5% variance (excellent estimation)

*Document Version: 1.0

*Last Updated: February 3, 2026

*Next Review: March1, 2026 (mid-project checkpoint)