Project Proposal: Methodology

Erika Bond

8 December 2025

Mrs. Carter

Technical Information

This section outlines the technical requirements, software, skills, and development timeline needed to build the direct-connection modeling platform. While the concept is ambitious, the technology required to create it already exists and is widely used in similar web-based marketplaces. The project will require a combination of front-end development, back-end infrastructure, database management, and user experience design, along with collaboration from technical advisors and creative professionals.

1. Technical Stack

The platform will be built using modern, widely supported web technologies to ensure scalability, security, and reliability.

Front-End (User Interface and Experience)

This is the part of the platform that users interact with directly.

- HTML5 for structure
- CSS3 for styling and responsive design
- JavaScript for interactions
- React.js for interface components and state management
- Tailwind CSS for fast, consistent UI styling

These tools are widely used for professional web platforms and are well-documented, which makes them accessible for learning and team collaboration.

Back-End (Server and Application Logic)

This controls user authentication, messaging, booking, and data processing.

- Node.js for server-side logic
- Express.js as the server framework
- REST API for communication between front-end and database

This stack is commonly used for real-time applications such as booking and messaging systems because of its speed and flexibility.

Database and Storage

- MongoDB for user profiles, bookings, reviews, and contracts
- Firebase Storage or AWS S3 for storing images and videos securely

Non-relational databases like MongoDB are ideal for handling large amounts of flexible profile data and media-heavy platforms.

Authentication and Security

- JWT Authentication for secure login sessions
- OAuth for optional login through Google or Apple
- Data encryption (HTTPS/SSL) for user protection

Security is essential due to the storage of contracts, images, and personal data.

Payment Processing

- Stripe API for handling secure payments, deposits, and payouts
- Supports direct model payment without agency interference
- Refunds and payment tracking included

Stripe is widely accepted in freelance and marketplace platforms and includes financial compliance features.

Messaging and Notifications

- Socket.io for real-time messaging
- Firebase Cloud Messaging for email and push notifications

These tools allow instant communication without relying on external apps.

Design and Prototyping Tools

- Figma and Canva for wireframes and UI design
- Adobe Photoshop and Lightroom for image optimization
- Canva for quick visual mockups

2. Team and External Support

This project requires a multidisciplinary team. While one individual can prototype the product, full-scale development would realistically involve:

- Front-End Developer to build the interface
- Back-End Developer to manage databases and APIs
- UX/UI Designer to design flows and usability
- Legal Advisor to review contracts and usage rights
- Payment Processing Advisor for financial compliance
- Creative Industry Consultant to ensure the platform fits real model and brand workflows

Some of these roles can be outsourced, supported by mentors, or fulfilled through academic partnerships.

3. Skills Required Existing Skills

- UX research and persona development
- Creative industry knowledge
- Branding and content strategy
- Project planning

Skills to Learn or Strengthen

- JavaScript and React development
- Database management with MongoDB
- API integration
- Secure authentication systems
- Payment API implementation
- Web accessibility and performance optimization

According to Stack Overflow's Developer Survey, these are among the most in-demand and supported technologies, making them realistic to learn within an academic or startup environment.

4. Development Timeline and Estimated Hours

Below is a realistic 6-month development timeline for a functional MVP (Minimum Viable Product). This assumes part-time development with a small team.

Phase Task Estimated Time Month 1 Research, feature finalization, UX wireframes 40–60 hours Month 2 UI design in Figma, brand identity, user flows 40–50 hours Month 3 Front-end build (login, profiles, dashboard) 70–90 hours Month 4 Back-end build (database, APIs,

messaging) 80–100 hours Month 5 Payment system, contracts, reviews, security 50–70 hours Month 6 Testing, bug fixes, beta launch, feedback 40–60 hours

Total Estimated Development Time: Approximately 320–430 hours for a working MVP.

Optional future development phases would include mobile app development, advanced analytics, and scaling infrastructure.

6. Technical Feasibility and Expert Support

All technologies listed are supported by extensive documentation from companies such as:

- Meta (React)
- MongoDB Documentation
- Stripe Developer Guides
- AWS Cloud Infrastructure

These tools are used in thousands of active platforms today. According to industry documentation and developer case studies, platforms with similar architecture successfully support millions of users with appropriate scaling.

This confirms that the platform is not only conceptually strong, but also technically realistic within both academic and startup environments.

7. Technical Summary

This product is technically feasible using modern, well-supported web technologies. It does not require experimental software or custom hardware. The platform can be built in structured phases using an agile development model and expanded over time. By combining front-end design, secure back-end systems, and cloud infrastructure, the platform can support real-world bookings, communication, contracts, and payments in a professional and scalable way.