Project Proposal (first draft)

Emelie Lustig, Miranda Teta, Manuella Tona Katureebe
Completed Monday June 17th 2024

Front-End Development:

1. HTML/CSS:
- Purpose: Structure and style web pages.
- Tools: HTML5, CSS3, Sass (for advanced styling).

2. JavaScript:
- Purpose: Add interactivity to the site.
- Tools: Vanilla JavaScript, modern frameworks/libraries like React.js, Vue.js, or Angular.

3. Responsive Design:
- Purpose: Ensure the website looks good on all devices.
- Tools: Bootstrap, Tailwind CSS.

Back-End Development:

1. Programming Languages:
- JavaScript (Node.js): Popular for building scalable web applications.
- Python: With frameworks like Django or Flask.
- Ruby: Using the Ruby on Rails framework.
- PHP: With frameworks like Laravel.

2. ServerSide Frameworks:
- Express.js (for Node.js)
- Django (for Python)
- Ruby on Rails (for Ruby)
- Laravel (for PHP)

Database Management:

1. SQL Databases:
- MySQL: Widely used and supported.
- PostgreSQL: Known for its advanced features and performance.

2. NoSQL Databases (if we prefer a non-relational database):
- MongoDB: Popular choice for flexible schema designs.

Authentication and Security:

1. Authentication:
- OAuth: For third-party authentication (e.g., Google, Facebook).
- JWT (JSON Web Tokens): For handling user sessions securely.

2. Security Best Practices:
- HTTPS: Secure the site with SSL certificates.
- Sanitization: Protect against SQL injection and XSS attacks.
- Password Hashing: Use libraries like bcrypt for secure password storage.

Reviews and Ratings:

1. Implementing Reviews:
- Database Schema: Design tables to store reviews and ratings.
- APIs: Create endpoints for submitting and retrieving reviews.
- User Interface: This allows users to write and read reviews easily.

Search and Filtering:

1. Elasticsearch: For advanced search capabilities.
2. Algolia: A hosted search engine that provides real-time search results.

Deployment and Hosting:

1. Web Servers:
- Nginx: High-performance web server.
- Apache: Another popular web server.

2. Hosting Platforms:
- AWS: Comprehensive cloud services.
- Heroku: Easy deployment for apps.
- DigitalOcean: Affordable cloud services.

3. Continuous Integration/Continuous Deployment (CI/CD):
- Jenkins: For automating the deployment process.
- GitHub Actions: For CI/CD workflows integrated with the repository.

Development Tools:

1. Version Control:
- Git: Essential for tracking changes and collaboration.
- GitHub/GitLab/Bitbucket: Platforms for hosting our repositories.

2. Code Editors/IDEs:
- Visual Studio Code: Popular and versatile code editor.
- PyCharm: For Python development.
- RubyMine: For Ruby on Rails development.

3. Task Runners and Build Tools:
- Webpack: Module bundler for JavaScript.
- Gulp/Grunt: Task runners for automating tasks.

Testing:

1. Unit Testing:
- Jest (JavaScript)
- PyTest (Python)
- RSpec (Ruby)
- PHPUnit (PHP)

2. EndtoEnd Testing:
- Cypress
- Selenium

Analytics and Monitoring:

1. Google Analytics: For tracking user behaviour.
2. Sentry: For error tracking and monitoring.

Soft Skills:

1. Project Management:
- Agile/Scrum: Methodologies for managing development.
- Tools: Jira, Trello, Asana.

2. Communication:

• 	Effective communication with team members and stakeholders.

People needed:

Core Team: 1218 people

- 1 Project Manager
- 23 FrontEnd Developers
- 23 BackEnd Developers
- 12 UI/UX Designers
- 1 Database Administrator
- 12 DevOps Engineers
- 2 QA Testers
- 1 Security Specialist
- 1 Marketing Specialist
- 12 Customer Support
- 1 Content Writer (optional)
- 1 Business Analyst (optional)

Potential revenue streams:

1. Listing Fees:
- Description: Charge landlords or property managers a fee to list their properties on the platform.
- Model: Can be a one-time fee per listing, a monthly subscription, or a tiered pricing model based on the number of listings or features (e.g., highlighted listings).

2. Service Fees:
- Description: Take a percentage of each rental transaction processed through the platform.
- Model: Similar to Airbnb, charge tenants a booking fee and/or landlords a service fee.

3. Subscription Plans:
- Description: Offer premium subscription plans to landlords for additional features.
- Features: Enhanced visibility, detailed analytics, featured listings, and access to premium customer support.

4. Advertising:
- OnSite Advertising: Sell ad space on the website to relevant businesses (e.g., moving companies, furniture stores, or insurance providers).
- Sponsored Listings: Offer landlords the option to pay for their properties to be featured prominently in search results.

5. Tenant Services:
- Utility Setup: Partner with utility companies to offer a service where tenants can set up utilities through the platform.
- Insurance: Partner with insurance companies to offer renters' insurance through the platform, taking a commission on each sale.

6. Affiliate Marketing:
- Description: Earn commissions by promoting third-party products and services relevant to the users.
- Examples: Furniture and home decor stores, moving services, and storage solutions.

7. Data Services:
- Market Insights: Sell anonymized data and insights to real estate developers, universities, and businesses interested in student housing trends.

- Research Reports: Produce and sell detailed reports on housing trends, rental prices, and market demand.

8. ValueAdded Services:
- Property Management Tools: Offer software tools for landlords to manage their properties, handle maintenance requests, and communicate with tenants.
- Legal Services: Partner with legal firms to offer landlords and tenants access to legal advice and services.

9. Events and Webinars:
- Description: Host events, workshops, and webinars on topics related to student housing, moving tips, and tenant rights.
- Revenue: Charge attendance fees or secure sponsorships for these events.

10. Freemium Model:
- Description: Provide basic features for free to attract users, and charge for premium features.
- Features: Advanced search filters, virtual tours, tenant background checks, and detailed property analytics.

How to start:

1. Define Our Vision and Goals:
- Clarify Our Mission: Clearly define what you aim to achieve with our platform.
- Set Goals: Identify short-term and long-term goals, such as launching an MVP (Minimum Viable Product), gaining a certain number of users, and securing funding.

2. Conduct Market Research:
- Understand the Market: Research the student housing market in Canada, identifying key pain points for students and landlords.
- Identify Competitors: Analyze platforms like Airbnb, Facebook Marketplace, and Craigslist to understand their strengths and weaknesses.
- Surveys and Interviews: Conduct surveys and interviews with potential users (students and landlords) to gather insights on their needs and preferences.

3. Develop a Business Plan:
- Revenue Model: Decide on how you will monetize the platform (e.g., listing fees, service fees, advertising).
- Budgeting: Outline our budget, including development costs, marketing expenses, and operational costs.
- Funding: Identify potential funding sources (e.g., bootstrapping, angel investors, venture capital).

4. Define the Product Requirements:
- Core Features: List the essential features for the MVP, such as user registration, property listings, search functionality, and reviews.
- Additional Features: Identify features to add in later phases, such as advanced search filters, premium listings, and integration with third-party services.

5. Assign Roles and Responsibilities:
- Economics Major: Focus on business strategy, market research, and financial planning.
- Psychology Major: Handle user experience research, risk analysis, and data analysis.
- Philosophy Major: Lead the initial development efforts, leveraging the programming skills in Python and R.

6. Learn Necessary Skills:

- FrontEnd Development: Consider learning basic HTML, CSS, and JavaScript, as well as a front-end framework like React or Vue.
- BackEnd Development: Learn a serverside language like Node.js, Python (Django or Flask), or Ruby on Rails.
- Database Management: Get familiar with SQL databases (e.g., PostgreSQL, MySQL) and NoSQL databases (e.g., MongoDB).
- Version Control: Use Git for version control and collaborate via platforms like GitHub.

7. Build an MVP (Minimum Viable Product):
- Prototyping: Create wireframes and mockups of the platform using tools like Figma or Adobe XD.
- Development: Start building the core functionality of the platform.
- FrontEnd: Develop the user interface and ensure it's responsive.
- BackEnd: Set up the server, database, and API endpoints.
- Testing: Conduct thorough testing to ensure functionality and fix any bugs.

8. Launch and Iterate:
- Beta Testing: Launch a beta version of the platform to a small group of users and gather feedback.
- Marketing: Promote the platform through social media, university networks, and online communities.
- Feedback Loop: Continuously gather user feedback and make improvements to the platform.

9. Scale and Grow:
- Expand Features: Gradually introduce additional features based on user feedback and market demand.
- Partnerships: Form partnerships with universities, student organizations, and housing providers.
- Funding and Monetization: Seek funding to scale the platform and implement the monetization strategies.

Tools and Resources:

- Learning Platforms: Use resources like CodeAcademy, Coursera, Udemy, and freeCodeCamp to learn necessary technical skills.
- Project Management: Use tools like Trello, Asana, or Jira to manage tasks and collaborate efficiently.
- Development Tools: Set up a development environment with code editors like Visual Studio Code, and GitHub for version control, and a cloud platform like AWS or Heroku for deployment.