Project Overview:

The purpose of this project is to create a user-friendly web platform that allows individuals to list and discover rental properties for short-term stays. This project aims to provide a seamless experience for hosts to showcase their properties and for guests to find accommodations that suit their preferences and needs.

Detailed Project Requirements:

1. Objectives for Student portal
   1. User Registration: An authentication system that allows buyers to create accounts, log in.
   2. Profile Management: Students can update their profiles, preferences, and search criteria.
   3. Property Search: Portal should be able to search property with high efficiency based on location, type, price range, and amenities, with a user interface.
   4. Personalised Recommendations: Based on buyer’s search history and present location, recommendations will be provided for the buyer to see nearby properties.
   5. Payment Gateway Integration: Once the desired property is decided, the user should be able to make payments for property reservations.
   6. Real-time Availability: The portal should be able to show real-time availability of a property to prevent double bookings.
   7. Saved Searches and Favourites: Users should be able to bookmark their desired properties (Similar to a cart) before making a purchase. This should also show the price difference between each property to make an informed decision by the user.
   8. Communication with Sellers and Booking updates: After completing the payment for their selected property, buyers should receive an invoice for their transaction. Additionally, they should be able to communicate with sellers, enabling them to ask questions.
   9. Notifications: Automated alerts for new listings, booking confirmations, and updates.
2. Objectives for Landlord Portal
   1. Account Creation: Landlords can register and create profiles for their properties.
   2. Property Management: Landlords can add, edit, or remove property listings with details such as images, amenities, and pricing.
   3. Booking Management: Real-time updates on booking requests, confirmations, and cancellations.
   4. Messaging and Communications: Communications between landlords and students through a messaging feature, allowing for questions, confirmation of deals.
3. Objectives for Admin Portal
   1. Content Management: Capability to update website content, banners, and announcements.
   2. User Management: The admin must be able to manage user accounts, including creating, updating, and deactivating accounts as needed.
   3. Transaction Monitoring: Implement a system to monitor and review payment transactions, ensuring secure and accurate processing.
   4. Property Listing Overview: The admin must be able to review property listings, ensuring they meet the standards and deactivate them if needed.
   5. User Support and Dispute Resolution: Equip the admin with a communication system to assist users in resolving any disputes or issues that may arise.

Technical Requirements (High-Level Overview):

Frontend:

* Next.js will be employed for crafting the user interface, customised to ensure optimal UI responsiveness. Responsive design will be supported for dimensions of minimum 800px by 1400px.
* Next.js will assist in implementing future SEO optimizations.
* UI libraries and frameworks include Tailwind CSS, Bootstrap, and Material UI.

Backend:

* Ruby on Rails, GraphQL will be utilised for swift API development which will facilitate communication between the frontend and backend.
* Targeted API’s will be highly efficient with an industry-standard response time of 150ms for improved frontend responsiveness.

Database:

* PostgreSQL will be employed for handling relational data.

Authentication:

* OAuth2 will be used for secure authentication.

Payment Gateway:

* PayPal and Razorpay will serve as payment integration portals.

Hosting and Deployment:

* AWS will host servers, and S3 buckets will store invoices and project documents.

Security:

* Adhere to best practices for web security tailored to Ruby on Rails, encompassing encryption and optimal coding practices.

Monitoring and User Analytics:

* Google Analytics and New Relic will be employed to monitor user behaviour and track user locations.

Student Portal Feature Estimation:

| Feature Description | Estimated Time |
| --- | --- |
| Initial Design Phase (Wireframes) | 24 hours |
| Main Screen (Actionable content, Static content) | 50 hours |
| User Registration, Login Screen (1.1) (Actionable content, Static content) | 24 hours |
| Profile Management Screen (1.2) (Actionable content, Static content) | 24 hours |
| Smart Search (1.3) (Actionable content, Static content) | 24 hours |
| Search Result Screen (Preview Grid) (Actionable content, Static content) | 50 hours |
| Room Description Screen (Actionable Content) | 24 hours |
| Sitemap (Static Content) | 16 hours |
| Contact Us (Static Content) | 16 hours |
| About Us (Static Content) | 16 hours |
| faq Page | 16 hours |
| Privacy Policy (Static Content) | 8 hours |
| News & Media (Static Content) | 8 hours |
| Booking Flow (Static Content) | 8 hours |
| Stories (Static Content) | 8 hours |
| Payment View/Success, Failure (Actionable Content) | 24 hours |
| Messaging Screen (Actionable Content) | 24 hours |
| Api Development from Scratch |  |
| Registration, Login Api | 8 hours |
| Fetching Data from DB to main Screen | 8 hours |
| Search Screen Api | 8 hours |
| Preview Tabs Api | 8 hours |
| Room description api | 8 hours |
| Payment Gateway Integration | 8 hours |
| Payments Api, Success, failure scenarios | 8 hours |
| Deployment AWS, hosting, DB setup | 8 hours |
| Total: | 428 hours ~ 2 months |

Landlord Portal Feature Estimation:

| Feature Estimation | Estimated Time |
| --- | --- |
| Initial Design Phase (Wireframes) | 16 hours |
| Main Screen (Actionable content, Static content) | 50 hours |
| User Registration, Login Screen (1.1) (Actionable content, Static content) | 50 hours |
| Profile Management Screen (1.2) (Actionable content, Static content) | 50 hours |
| Property’s View Page | 40 hours |
| Property Management Edit Page | 40 hours |
| Messaging Page | 40 hours |
| Transaction’s, Wallet Page | 40 hours |
| Api Development From Scratch |  |
| Registration/Login Api’s | 8 hours |
| Fetch Data from DB to main screen | 8 hours |
| Property Listing Api | 8 hours |
| Edit Page Api | 8 hours |
| Wallet’s Api | 8 hours |
| Real-Time Update Api | 8 hours |
| Messaging Screen Api’s | 8 hours |
| AWS Deployment, DB Setup, Hosting | 8 hours |
| Total: | 390 hours ~ 2 months |

Admin Portal Feature Estimation:

| Feature Estimation: | Estimated Time |
| --- | --- |
| InitialDesign Phase (Wireframes) | 16 hours |
| Transaction’s Page | 40 hours |
| Support Requests Page | 40 hours |
| Support Messaging Page | 40 hours |
| Landlord List’s Page View/Edit | 40 hours |
| Property Listing’s Edit Page | 40 hours |
| Wallet’s List Page | 40 hours |
| Api Development From Scratch |  |
| Main Screen Api | 24 hours |
| Support Page Api’s | 16 hours |
| Landlord Lists Api | 16 hours |
| Property Listing Page | 24 hours |
| Wallets List Page | 24 hours |
| Total: | 360 hours ~ 2 months |

Cost Estimations:

10$ per hour

Estimated Total hours: ~ 1100 - 1200 hours

Estimated Project Cost: 12000$ ~ 10 lakh rupees

TimeLine:

The project will be completed in a 6-month timeline. The Student Portal will be finished in 2 months. The next 2 months will be for iterations of the Landlord Portal, and the following 2 months will be dedicated to the Admin Portal.

Agile methodology will be followed, with a continuous review process after completing each screen. The estimated cost of the whole project will be around 10 lakh rupees.