Gharpayy PRODUCT Plan – Subhash Jadav

1. Vision – What will Gharpayy solve?

Gharpayy will simplify and digitize the painful process of finding PGs for students and working professionals by creating a reliable, fast, and transparent rental platform. Think Airbnb, but for Indian PGs — with verified listings, clear rules, and smooth bookings.

2. Core MVP Features

Tenants (Users):

- Sign up / Login (Phone/Email + Firebase Auth)
- Home screen with PG suggestions (based on location or filters)
- PG listing details: photos, price, amenities, reviews, map, rules
- Contact PG owner or request a call back
- Easy booking and online payment
- Terms & conditions / agreement visibility

PG Owners (Hosts):

- Add new PG listing (form with images, price, location, rules)
- Manage listings (edit, delete, update availability)
- Respond to inquiries and bookings
- View booking history / transactions

Admin Panel:

- Approve/reject PG listings
- Monitor flagged content or reviews
- Manage users and verify PG owners
- Analytics dashboard (optional in later version)

3. Tech Stack

Frontend:

- **Android App:** Kotlin + Jetpack Compose (for latest scalable UI)
- (Optional later) **Kotlin Multiplatform** to support iOS/Web if needed

Backend:

- **Firebase** (Auth, Firestore, Storage, Cloud Functions) for rapid development and real-time sync

Database:

- **Cloud Firestore** – scalable, real-time database

Hosting (Admin Panel / Web support):

- Firebase Hosting or Vercel (if we create a web admin)

Maps Integration:

- **Google Maps SDK + Places API** – for PG location and autocomplete search

Payments:

- **Razorpay** or **PhonePe** Payment Gateway (UPI + card)

Other Tools:

- Crashlytics, Firebase Analytics, Lottie for animations, Glide for image loading

---

4. 2-Week Sprint Plan (Execution Strategy)

Sprint Goal: Build a working MVP prototype of the tenant app + backend (Firebase) + basic owner flow

Day	Module	Task
1–2	Planning & Setup	Project repo setup, Firebase config, toolstack finalization
3–5	Authentication	Phone/email login with Firebase, onboarding screens
6–8	Home & PG Listing	Home screen, fetch & show PG listings from Firestore
9–10	PG Detail & Booking	PG detail screen, rules, image gallery, contact owner
11–12	Booking & Payments	Integrate Razorpay/PhonePe sandbox
13	Owner Add Listing	Simple form to add a PG (title, location, price, amenities)
14	Testing + Polish	Bug fixes, design polish, documentation

---

5. Team Setup

Preferred: Small squad of 2–3 members for speed and clarity.

- **Frontend & Firebase Auth (Me):** Kotlin + Jetpack Compose + Firestore

- **Backend Lead:** Handles Cloud Functions, payment flow, Firestore rules
- **Design/PM (Optional):** UI polish, testing, task tracking (Not mandatory for MVP)

If solo, I'll prioritize the tenant flow and keep owner/admin features minimal but functional.

---

6. Dependencies (From Gharpayy Team)

To build faster, I'd need:

- **Branding Kit:** Logo, colours, basic UI direction (Figma or moodboard)
- **Sample PG Data:** Some dummy listings for testing/demo
- **Razor pay credentials:** Sandbox keys for testing
- **Clarity on contract/legal feature:** If Gharpayy plans to support digital agreements
- **Quick feedback loop:** Slack/Discord group to unblock doubts quickly

---

7. Final Output after Sprint 1

- Fully working Android MVP for Tenants:
  - Login
  - Home + Listings
  - PG details page
  - Booking and payment test flow
- Firebase backend + Firestore data schema
- PG owner can upload a PG via simple form
- Brief documentation for handoff or team expansion

---

8. AI + Tools Strategy

AI Tools I'll use:

- **Figma AI:** For UI suggestions, auto-layouts
- **ChatGPT :** For logic, Firebase rules, debugging faster
- **Cursor IDE:** AI code refactoring, faster component creation
- **Zapier (optional later):** To automate booking alerts or emails
- **Notion or Trello:** For sprint/task management

Where AI helps:

- Writing boilerplate (e.g. Firebase listeners, error handling)
- Generating test data
- Debugging Firebase rules or backend logic
- Designing faster UI mockups

Manual Work: Booking flow, UI polish, testing, user experience — to make it *feel* like a real product, not just a demo.

On Going Project Details

E Commerce App

Developed an e-commerce application to practice implementing user authentication with Firebase and to understand the fundamentals of Firestore database integration. Additionally, gained hands-on experience in integrating Firebase services within a Jetpack Compose-based Android application.

🔗 GitHub: https://github.com/subhash-jadav/E-CommerceApp/tree/master

Social Polling App *(Ongoing)*
Developing a modern polling application using **Firebase** and **Jetpack Compose**, enabling users to create, vote, and interact on polls. Implemented **real-time voting**, **infinite scrolling**, and **Google/OTP authentication** to enhance user experience.

Status : On Going