PROJECT ABSTRACT

PG Roommate Finder with Lifestyle Matching

Finding a paying-guest roommate in a new city is luck of the draw — most students pick from broker listings and discover the sleep-schedule clash after moving in. This project is RoomSync, a PG roommate finder that ranks candidates by lifestyle compatibility: a 12-question quiz captures sleep, food, smoking, noise and budget habits, and each match shows a scored breakdown so the decision is informed, not random. Suitable for B.E./B.Tech final-year projects in Computer Science, IT and Web Development.

Project type: Software (web application)

Availability: Built to order

Suitable branches: Computer / IT, Web Development

Technologies: HTML5, CSS3, JavaScript ES6, localStorage, Matching algorithm, SVG components

Price: Request a quotation

Prepared by Projectech · projectech.in — for academic project reference.

1. Title

PG Roommate Finder with Lifestyle Matching — a built-to-order software project delivered with a working web application, complete source code, project report, presentation and viva preparation kit.

2. Introduction / Background

Students relocating to cities like Pune for college or a first job face the same ritual: broker WhatsApp forwards, rushed flat visits, and a roommate chosen on gut feel. The expensive failures are lifestyle mismatches — a night owl paired with an early riser, a smoker with someone who cannot stand it, incompatible food habits in a shared kitchen. Existing listing sites show rooms, not people, and profile-based apps leave compatibility to chat vibes. This project builds RoomSync, a roommate-matching web app centered on a lifestyle quiz. Twelve questions map answers to factor scores across sleep, cleanliness, food, smoking, guests, noise, work routine and budget. Core lifestyle factors carry double weight, preferences single weight, and any deal-breaker mismatch caps the total — so a 92 genuinely means day-to-day compatible. Browse views rank candidates by score with filters; each profile shows a factor-by-factor compatibility breakdown with the honest soft mismatches called out.

3. Problem Statement

Students relocating to cities like Pune for college or a first job face the same ritual: broker WhatsApp forwards, rushed flat visits, and a roommate chosen on gut feel. The expensive failures are lifestyle mismatches — a night owl paired with an early riser, a smoker with someone who cannot stand it, incompatible food habits in a shared kitchen. Existing listing sites show rooms, not people, and profile-based apps leave compatibility to chat vibes. This project builds RoomSync, a roommate-matching web app centered on a lifestyle quiz. Twelve questions map answers to factor scores across sleep, cleanliness, food, smoking, guests, noise, work routine and budget. Core lifestyle factors carry double weight, preferences single weight, and any deal-breaker mismatch caps the total — so a 92 genuinely means day-to-day compatible. Browse views rank candidates by score with filters; each profile shows a factor-by-factor compatibility breakdown with the honest soft mismatches called out.

4. Objectives

- **12-question lifestyle quiz** — Sleep, cleanliness, food, smoking, guests, noise, work routine and budget are captured in a guided quiz with a progress tracker and a documented scoring method.
- **Weighted compatibility scoring** — Core factors (sleep, smoking, food) carry double weight, preferences single; any deal-breaker mismatch caps the total at 60, so high scores are trustworthy.
- **Ranked match browsing** — Candidates are ranked by lifestyle compatibility with match-score rings, filterable by budget, diet, smoking, sleep schedule, WFH-friendliness and area.
- **Factor-by-factor breakdown** — Each profile shows per-factor scores with plain-language explanations (both early risers, 11 PM – 6:30 AM) and a deal-breaker check with soft mismatches named.
- **My listing management** — Users publish their own listing with area, budget range and move-in date, and see their match-profile completeness driving visibility.
- **Verified-profile chat** — Connect requests and chat are restricted to ID-verified, college-email-verified profiles with mutual connections shown, keeping the social layer safer.

5. Proposed Methodology

The project is delivered built-to-order in four stages. (1) **Requirements & data model:** the domain entities (roommate finder workflows) are modeled with their states and transitions. (2) **Application build:** the single-file web app is implemented with the dashboard, workflow and reporting views verified against the seeded dataset. (3) **Rules & validation:** business rules (thresholds, approvals, escalation) are encoded and tested against edge cases. (4) **Documentation:** report, PPT, viva Q&A; and user manual are prepared to match the delivered build.

6. System Architecture / Working

The application follows a client-side architecture: a single HTML shell, a CSS design system, and a JavaScript application layer with separate modules for data (localStorage persistence), views (dashboard, workflow, reports) and rules (the k-1604 business logic). A new user completes the 12-question lifestyle quiz; each answer maps to a 0–100 factor score across the lifestyle dimensions. Factor scores are weighted — core lifestyle factors double, preferences single — and any deal-breaker mismatch (e.g. smoking) caps the total at 60. The user publishes a listing with area, budget range and move-in date; the match profile completeness percentage drives how prominently it ranks.

7. Hardware / Software Requirements

- Any modern laptop/desktop with an evergreen browser (Chrome, Edge, Firefox) — design target
- No server, database install or build toolchain required for the demo build
- HTML5, CSS3, vanilla JavaScript (ES6); browser localStorage for persistence
- Single-file deployment: the app runs offline after download
- Optional: static hosting for shared access; backend extension for multi-user use (configurable)

8. Expected Outcomes

- A working web application implementing the full documented workflow with realistic seeded data.
- Dashboard, workflow and reporting views verified against the demo dataset.
- Business rules (thresholds, approvals, escalation) encoded and demonstrable.
- A complete student kit: source code, report, PPT, viva Q&A;, user manual and deployment guide.

9. Applications

- Academic final-year project in Computer Science, IT or Web Development programs.
- Direct use by the target users (pg matching workflows) after replacing demo data.
- Teaching aid for web-application design, state management and dashboard development.

10. Future Scope

- Multi-user backend with authentication and role-based access (configurable extension).
- Real notification integrations (email/SMS gateway) for alerts and reminders.
- File-upload support for attachments (bill scans, documents) with cloud storage.
- Mobile-responsive refinements and a companion mobile view.