🏠 StayNest – Full Stack Web Application

Project Report

Author: Sakethram Reddy Palugulla
Technologies: React.js, PHP, MySQL, Bootstrap, Apache (XAMPP)

---

1. Introduction

StayNest is a full-stack hybrid web application that helps students and professionals find affordable, comfortable, and verified accommodations such as PGs, hostels, and co-living spaces across major Indian cities. The platform is designed to simplify the accommodation search process by combining a modern, interactive React.js frontend with a stable and secure PHP–MySQL backend. It offers a smooth user experience where users can search for properties by city, sign up or log in to their accounts, view property details, and mark their favorite listings.

---

2. Objectives

The main objective of StayNest is to provide an intuitive and responsive web-based solution for student housing discovery. It aims to reduce the friction of finding hostels or PG accommodations by bringing all available options to a single unified platform. The application enables users to:

- Search for PGs and hostels based on city and preference filters.
- Sign up, log in, and manage their personal profiles.
- Mark properties as “interested” to save them for later viewing.
- Access a dashboard showing all saved and interested listings.
- Experience a visually appealing and responsive user interface.

The project’s long-term goal is to evolve into a scalable full-stack solution with support for real-time data, reviews, and interactive maps.

---

3. System Architecture

StayNest follows a **three-tier architecture**, comprising the frontend (React.js), backend (PHP APIs), and database (MySQL).

1. **Frontend (React.js):**
   The React frontend, located in the react-app/ folder, handles the user interface, routing, and dynamic content updates. It interacts with the backend APIs through Axios requests. After development, the React application is built using npm run build, and the generated build folder is placed in the main PHP directory to run locally via Apache.

2. **Backend (PHP):**
   The PHP backend acts as the bridge between the React UI and the MySQL database. It exposes REST-style APIs for authentication, property retrieval, and user interests. The backend handles form submissions, session management, and validation using secure PHP practices.

3. **Database (MySQL):**
   MySQL serves as the persistent data layer for storing user accounts, property listings, city data, and user interests. It ensures relational consistency and quick query retrieval using indexed tables.

Folder Structure Overview:

StayNest/

├── react-app/ # React frontend source code

├── api/ # PHP API endpoints

├── css/, img/, js/ # Static resources

├── build/ # React production build

├── index.php # Root PHP file serving React build

├── dashboard.php # User dashboard (React-integrated)

├── staynest.sql # Database schema

└── README.md

---

4. Technologies Used

StayNest combines modern web technologies to ensure performance, maintainability, and scalability.

Layer	Technology	Description
Frontend	React.js, Bootstrap, CSS3, HTML5	Provides an interactive and responsive UI
Backend	PHP 8.0	Handles authentication, sessions, and API responses
Database	MySQL	Stores users, properties, and relationships
Server	Apache (via XAMPP)	Hosts backend and serves frontend
Tools	Node.js, npm, phpMyAdmin, GitHub	Used for build, deployment, and version control

---

5. Key Features

StayNest provides a rich set of features that enhance both usability and functionality.

- **City-Based Search:** Users can search for available PGs and hostels by selecting a city.
- **User Authentication:** Secure signup and login features with password encryption.
- **Property Listings:** Dynamic property data displayed from the database with filters for price and gender.
- **Favorites System:** Users can mark properties as “Interested” and view them later in their dashboard.
- **Dashboard:** Personalized dashboard displaying user information and saved PGs.
- **Responsive Design:** The React-based UI adapts seamlessly to all screen sizes.
- **Session Management:** PHP session tracking ensures secure and persistent logins.

---

6. Database Design

The StayNest database schema is normalized and designed for scalability.

Table	Description
users	Stores user details (id, name, email, phone, password)
properties	Stores details about each property (name, rent, city, gender, description)
cities	Stores available cities and their metadata
interests	Maps users to their saved properties

This structure allows efficient queries, minimal redundancy, and a foundation for extending the application to include ratings, maps, and reviews in future iterations.

---

7. Workflow

The user flow begins when the user visits the StayNest homepage. The React frontend fetches data from the PHP backend via API calls. Unauthenticated users can browse properties by city, but to save properties or view dashboards, users must log in or sign up. Once authenticated, PHP maintains the user's session, allowing the React interface to display personalized data through dashboard components. The backend interacts with the database to fetch or store data and returns JSON responses to the React frontend, ensuring a dynamic and seamless experience.

---

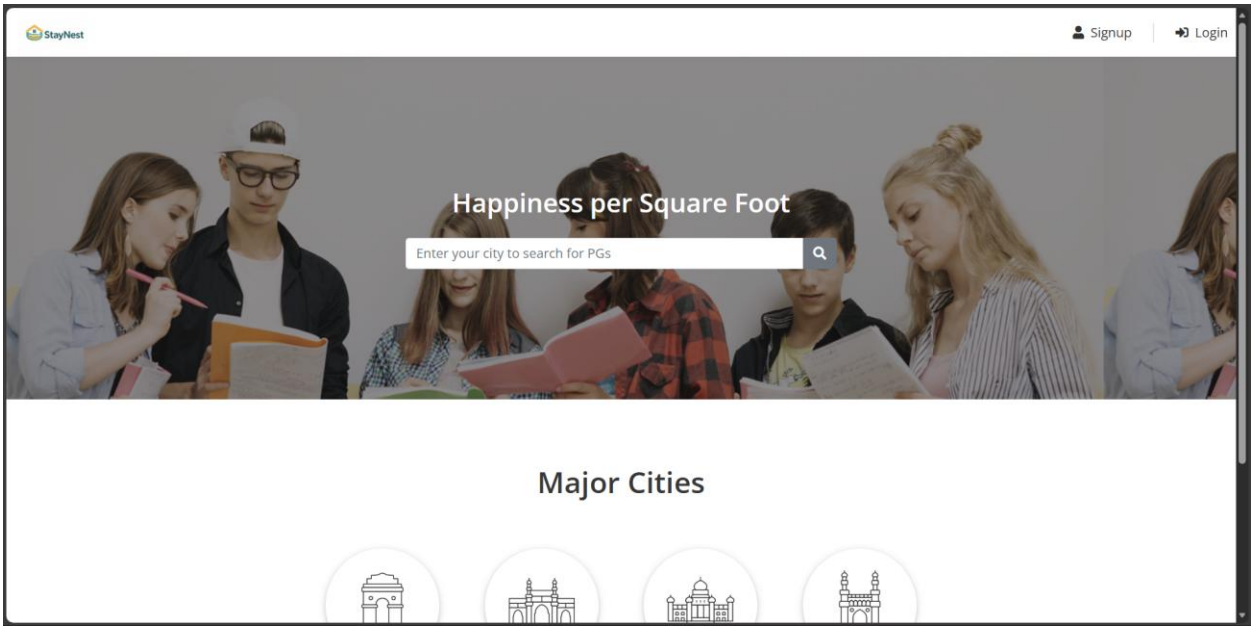
8. Implementation Details

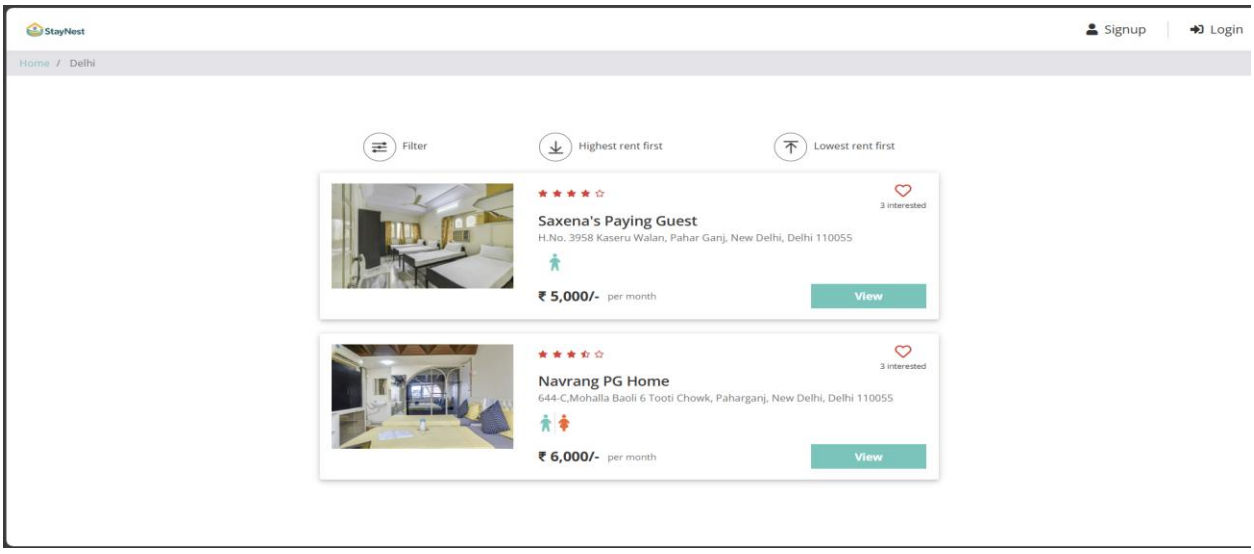
- **Frontend:** The React frontend utilizes reusable components for navigation bars, search sections, property cards, and modals. State management is achieved using React Hooks (useState, useEffect), and API calls are handled using Axios. Bootstrap ensures responsive styling.
- **Backend:** PHP scripts in the api/ directory act as endpoints for CRUD operations. Data is retrieved and formatted as JSON for the frontend. Security measures such as input sanitization and password hashing (password_hash() and password_verify()) are implemented.
- **Database:** MySQL tables are connected via foreign keys, and queries are optimized for property retrieval based on filters.

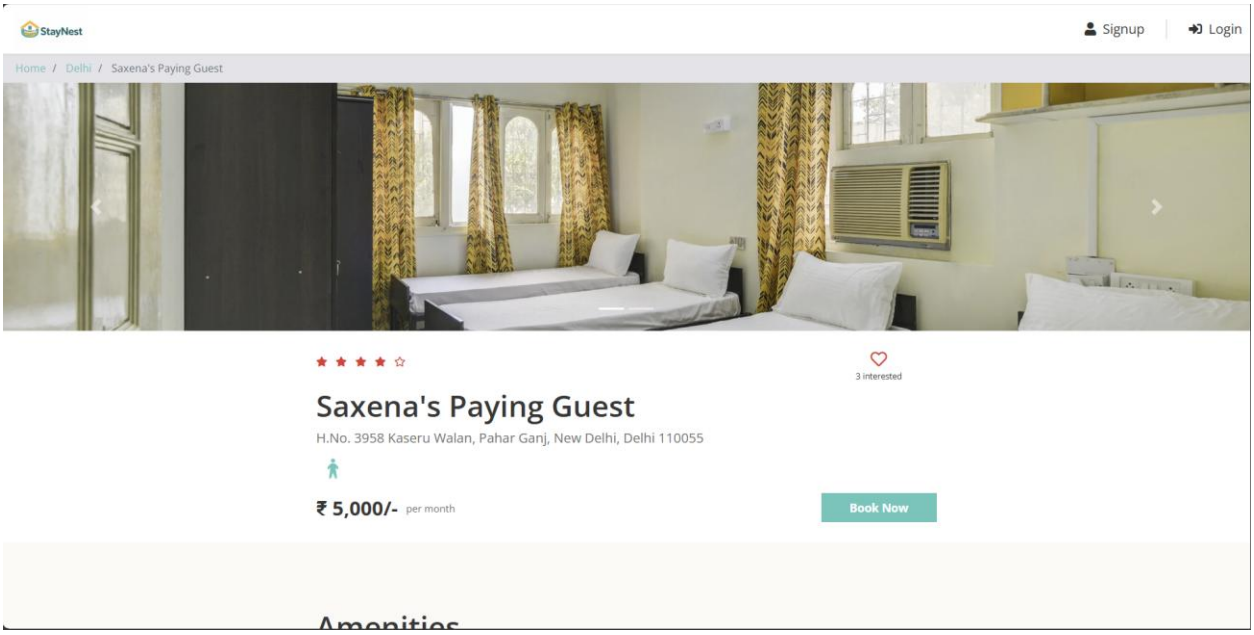
---

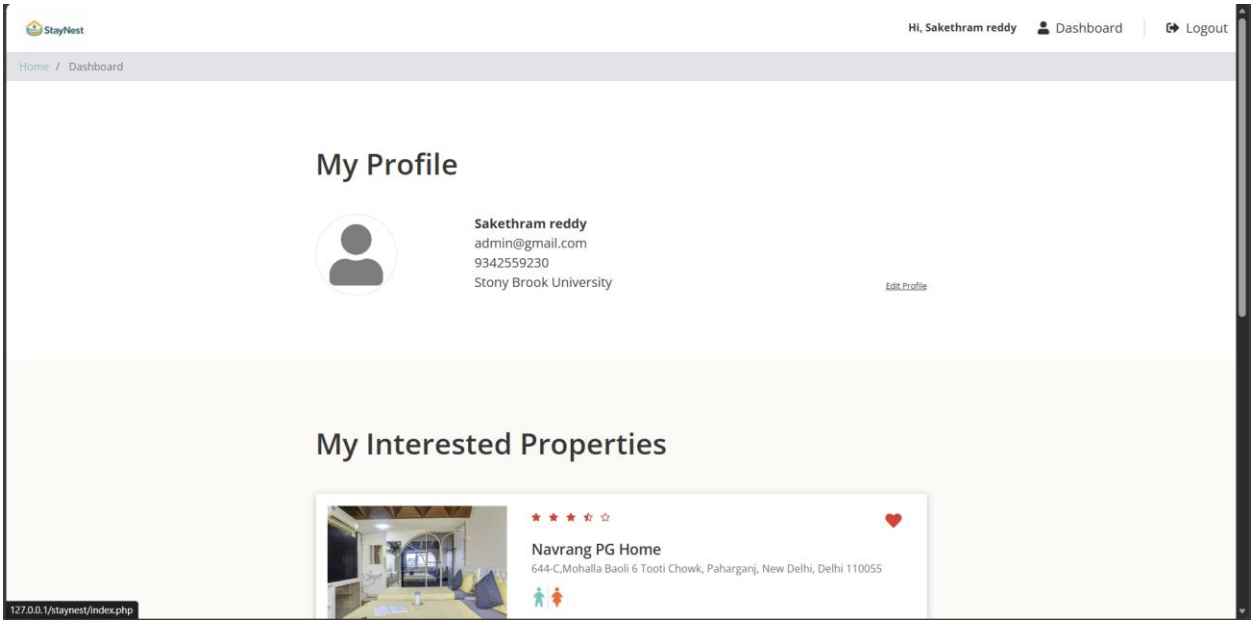
9. Results

The final StayNest system successfully integrates a React-based UI with a PHP–MySQL backend. It delivers smooth performance, clean navigation, and consistent responsiveness across devices. The new branding and logo give the project a professional identity while retaining the core purpose of PG Life. The system is fully functional in local environments via XAMPP and ready for cloud deployment.









---

10. Future Enhancements

Planned improvements for StayNest include:

- Integration of Google Maps API for geographic visualization of properties.
- Implementation of reviews and ratings for user feedback.
- Real-time chat between tenants and property owners.

- Admin dashboard for property management and verification.
- Full migration of backend APIs to Node.js or Django REST Framework.

---

11. Conclusion

StayNest successfully modernizes the PG Life platform using a full-stack architecture that combines React's dynamic frontend capabilities with PHP's stability and MySQL's reliability. It enhances the user experience with a clean UI, fast performance, and an intuitive workflow. This project demonstrates the integration of modern and legacy web technologies into a unified, scalable system that provides real-world value for students and professionals seeking accommodation.

Submitted by:
Sakethram Reddy Palugulla