

# Design SnapShot

## 1. Architecture Overview

Our system uses a modern, scalable 3-tier architecture.

- **Frontend (Next.js):** A React framework for building a fast, server-rendered web application that is responsive on all devices.
- **Backend (Java Spring Boot):** An enterprise-grade Java framework for our secure REST API, user authentication, and business logic.
- **Database (MySQL):** A reliable, open-source relational database to ensure data integrity for all users, vendors, and bookings.

## 2. Core Data Schema (MySQL)

Here is a simplified schema for our core MVP features:

```
-- Stores user information
CREATE TABLE Users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  email VARCHAR(255) NOT NULL UNIQUE,
  password_hash VARCHAR(255) NOT NULL,
  full_name VARCHAR(100),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Stores verified hotel listings
CREATE TABLE Hotels (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  location VARCHAR(100), -- (e.g., 'Pokhara', 'Kathmandu')
  description TEXT,
  is_verified BOOLEAN DEFAULT false,
  latitude DECIMAL(10, 8),
  longitude DECIMAL(11, 8),
```

```

    image_url VARCHAR(255)
);

-- Stores booking requests
CREATE TABLE Bookings (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    hotel_id INT,
    check_in_date DATE NOT NULL,
    check_out_date DATE NOT NULL,
    status VARCHAR(50) DEFAULT 'pending', -- (pending, confirmed, cancelled)
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(id),
    FOREIGN KEY (hotel_id) REFERENCES Hotels(id)
);

```

### 3. API Contract Outline

Our backend provides a simple RESTful API for the frontend.

- `POST /api/auth/register` - Creates a new user.
- `POST /api/auth/login` - Authenticates a user and returns a token.
- `GET /api/hotels` - Fetches all hotels.
- `GET /api/hotels?location=pokhara` - Fetches hotels filtered by location.
- `GET /api/hotels/{id}` - Fetches details for a single hotel.
- `POST /api/bookings` - (Auth required) Creates a new booking request.
- `GET /api/bookings/me` - (Auth required) Fetches all bookings for the current user.