

# Phase 1: Mock Core Data Setup for Real Estate Portal

This document contains everything needed to implement **Phase 1** of your portal: creating realistic mock data for testing Projects, Stands, Agents, Clients, and Reservations before integrating with the Core System. It includes migrations, models, controllers, views, seeds, and basic namespaces.

---

## 1. Migrations

### Create Users Table

```
rails generate migration CreateUsers id:uuid email:string username:string
password_digest:string phone_number:string role:integer status:integer
two_factor_enabled:boolean last_login_at:datetime failed_attempts:integer
```

Migration:

```
class CreateUsers < ActiveRecord::Migration[7.0]
  def change
    create_table :users, id: :uuid do |t|
      t.string :email, null: false, index: { unique: true }
      t.string :username
      t.string :password_digest
      t.string :phone_number
      t.integer :role, null: false, default: 0
      t.integer :status, default: 0
      t.boolean :two_factor_enabled, default: false
      t.datetime :last_login_at
      t.integer :failed_attempts, default: 0

      t.timestamps
    end
  end
end
```

### Create Agents, Clients, Admins

```
rails generate migration CreateAgents user:references full_name:string
agency_name:string license_number:string commission_rate:decimal status:integer
onboarded_at:datetime core_system_id:string
```

```
rails generate migration CreateClients user:references agent:references
full_name:string national_id:string email:string phone_number:string
date_of_birth:date address:text kyc_verified:boolean onboarding_source:integer
core_system_id:string
rails generate migration CreateAdmins user:references full_name:string
role:integer department:string
```

## Create Projects and Stands

```
rails generate migration CreateProjects core_project_id:string name:string
location:string description:text developer_name:string total_stands:integer
status:integer map_image_url:string
rails generate migration CreateStands core_stand_id:string project:references
stand_number:string area_sqm:decimal price:decimal status:integer
reserved_at:datetime sold_at:datetime latitude:decimal longitude:decimal
updated_from_core_at:datetime
```

## Create Reservations and Applications

```
rails generate migration CreateApplications client:references agent:references
stand:references application_number:string status:integer deposit_amount:decimal
rails generate migration CreateReservations client:references agent:references
stand:references application:references reservation_date:date expiry_date:date
status:integer total_price:decimal deposit_paid:decimal
remaining_balance:decimal core_reservation_id:string agreement_id:uuid
```

## Other Phase 1 Mocks (Optional)

- KYC Documents, Accounts, Transactions, Payments, Queries can be added with similar migrations if you want richer test data.

---

## 2. Models

### User Model

```
class User < ApplicationRecord
  has_secure_password

  enum role: { agent: 0, client: 1, admin: 2 }
  enum status: { active: 0, inactive: 1, suspended: 2 }

  has_one :agent
```

```
  has_one :client
  has_one :admin
  has_many :audit_logs
end
```

## Agent Model

```
class Agent < ApplicationRecord
  belongs_to :user
  has_many :clients
  has_many :reservations
end
```

## Client Model

```
class Client < ApplicationRecord
  belongs_to :user
  belongs_to :agent, optional: true
  has_many :applications
  has_many :reservations
end
```

## Project & Stand Models

```
class Project < ApplicationRecord
  has_many :stands
end

class Stand < ApplicationRecord
  belongs_to :project
  has_many :reservations
end
```

## Application & Reservation Models

```
class Application < ApplicationRecord
  belongs_to :client
  belongs_to :agent
  belongs_to :stand
  has_one :reservation
end

class Reservation < ApplicationRecord
```

```
belongs_to :client
belongs_to :agent
belongs_to :stand
belongs_to :application, optional: true
end
```

---

### 3. Namespaced Controllers (Admin Example)

```
rails generate controller admin_portal/projects
rails generate controller admin_portal/stands
```

Controller snippets:

```
module AdminPortal
  class ProjectsController < ApplicationController
    def index
      @projects = Project.all
    end

    def show
      @project = Project.find(params[:id])
      @stands = @project.stands
    end
  end
end
```

---

### 4. Views

- `admin_portal/projects/index.html.erb` → List projects.
- `admin_portal/projects/show.html.erb` → Show project details and its stands.
- `admin_portal/stands/_stand.html.erb` → Partial to render stand card.

---

### 5. Routes

```
namespace :admin_portal do
  resources :projects do
    resources :stands, only: [:index, :show]
  end
end
```

```
end  
end
```

## 6. Seeds (Phase 1 Mock Data)

```
# db/seeds.rb  
# Create 2 Agents  
agents = 2.times.map do |i|  
  Agent.create!(user: User.create!(email: "agent#{i}@test.com", password:  
  'password', role: :agent, status: :active), full_name: "Agent #{i}")  
end  
  
# Create 5 Clients each  
agents.each do |agent|  
  5.times do |i|  
    Client.create!(user: User.create!(email: "client#{i}@test.com", password:  
    'password', role: :client, status: :active), full_name: "Client #{i}", agent:  
    agent)  
  end  
end  
  
# Create 2 Projects with 10 Stands each  
2.times do |i|  
  project = Project.create!(name: "Project #{i}", location: "City #{i}",  
  status: :active, total_stands: 10)  
  10.times do |j|  
    Stand.create!(project: project, stand_number: "S#{j+1}", area_sqm: 100+j*5,  
    price: 50000 + j*2000, status: :available)  
  end  
end
```

## 7. Next Steps

- Phase 1 complete: all mock data ready.
- Run: `rails db:migrate && rails db:seed`
- Verify: Admin portal shows projects and stands correctly.
- Phase 2: start implementing **Agent and Client Portals** with dashboards, reservations, and payments.

This provides a **fully functional Phase 1** mock setup for early development, testing, and UI iteration before Core System integration.