# लाराभेल प्रोजेक्ट संरचना समीक्षा

## प्रमुख फोल्डरहरू र तिनीहरूको उद्देश्य

1. \*\*app/Http/Controllers\*\*:

- तीन प्रकारका रोलहरूका लागि कन्ट्रोलरहरू: Admin, Owner (होस्टल प्रबन्धक), र Student

- Admin: Dashboard, Contacts, Galleries, Hostels, Impersonation, MealMenu कन्ट्रोलरहरू

- Owner: Dashboard, Contacts, Galleries, Hostel, MealMenus, Payments, Reviews, Rooms, Students कन्ट्रोलरहरू

- Student: Dashboard, Profile, Room, Search, Student कन्ट्रोलरहरू

- साझा कन्ट्रोलरहरू: Onboarding, Payment, Registration, Role, Subscription

2. \*\*Models\*\*:

- Booking, College, Contact, Course, Gallery, Hostel, HostelImage, Meal, MealMenu, Payment, Plan, Review, Room, Student, Subscription, User आदि मोडेलहरू उपलब्ध

- Role-based प्रणाली: Role, Organization, OrganizationUser मोडेलहरू

- Onboarding प्रक्रियाको लागि: OnboardingProgress मोडेल

3. \*\*Requests\*\*:

- StoreHostelRequest, UpdateHostelRequest, StoreRoomRequest, UpdateRoomRequest, StoreStudentRequest, UpdateStudentRequest जस्ता फर्म रिक्वेस्टहरू

- ProfileUpdateRequest, ReviewRequest, ContactRequest आदि

4. \*\*Middleware\*\*:

- Admin.php, HostelOwner.php, CheckRole.php, RoleMiddleware.php, EnsureSubscriptionActive.php जस्ता मिडलवेयरहरू

- यी मिडलवेयरहरूले रोल-आधारित पहुँच नियन्त्रण गर्छन्

## अवस्थित सुविधाहरू/मोड्युलहरू

1. \*\*छात्र प्रबन्धन\*\*:

- StudentController, Student मोडेल

- StoreStudentRequest, UpdateStudentRequest

- ProfileController, ProfileUpdateRequest

2. \*\*होस्टल प्रबन्धन\*\*:

- HostelController (Admin), Hostel मोडेल

- HostelImage मोडेल

- StoreHostelRequest, UpdateHostelRequest

3. \*\*कोठा प्रबन्धन\*\*:

- RoomController, RoomsController (Owner)

- Room मोडेल

- StoreRoomRequest, UpdateRoomRequest

4. \*\*खाना प्रबन्धन\*\*:

- MealMenuController (Admin), MealMenusController (Owner)

- Meal, MealMenu मोडेलहरू

5. \*\*भुक्तानी प्रणाली\*\*:

- PaymentController, PaymentsController (Owner)

- Payment, Subscription, Plan मोडेलहरू

6. \*\*ग्यालेरी प्रणाली\*\*:

- GalleriesController (Admin र Owner)

- Gallery मोडेल

7. \*\*समीक्षा प्रणाली\*\*:

- ReviewController, ReviewsController (Owner)

- Review मोडेल

8. \*\*प्रयोगकर्ता भूमिका प्रणाली\*\*:

- RoleController, Role मोडेल

- ImpersonationController (Adminले अन्य प्रयोगकर्ताको रूपमा प्रवेश गर्न सक्छ)

- Organization, OrganizationUser मोडेलहरू

9. \*\*अन्य सुविधाहरू\*\*:

- Onboarding प्रक्रिया (OnboardingController, OnboardingProgress)

- सन्देश पठाउने (ContactRequest, ContactFormSubmitted मेल)

- सर्च प्रणाली (SearchController)

## अपूर्ण वा अनुपस्थित भागहरू

1. \*\*असंगत कन्ट्रोलर संरचना\*\*:

- RoomController.php (मुख्य) र RoomsController.php (Owner) जस्ता दोहोरिएका कन्ट्रोलरहरू

- GalleriesController, PaymentsController, ReviewsController लागि पनि यही समस्या

2. \*\*रोल-आधारित पहुँच नियन्त्रण\*\*:

- CheckRole.php र RoleMiddleware.php छन् तर सबै कन्ट्रोलरमा लागू भएको हो कि छैन भन्ने स्पष्ट छैन

3. \*\*भ्यु संरचना\*\*:

- भ्यु फोल्डर संरचना देखिँदैन, जुन भूमिका अनुसार (admin, owner, student) हुनुपर्छ

4. \*\*API र Web कन्ट्रोलर\*\*:

- controller.api.stub, controller.model.api.stub जस्ता स्टबहरू छन् तर कति कन्ट्रोलरहरूले API समर्थन गर्छन् भन्ने स्पष्ट छैन

5. \*\*फर्म रिक्वेस्ट प्रमाणीकरण\*\*:

- StoreHostelRequest, UpdateRoomRequest आदि छन् तर तिनीहरूमा प्रमाणीकरण नियमहरू पूरा गरिएको हो कि छैन भन्ने अस्पष्ट छ

## भूमिका (Roles) कार्यान्वयन

1. \*\*प्रशासक (Admin)\*\*:

- सबै होस्टल, प्रयोगकर्ता र सिस्टम सेटिङ्हरू व्यवस्थापन गर्न सक्छ

- ImpersonationController मार्फत अन्य प्रयोगकर्ताको रूपमा प्रवेश गर्न सक्छ

- Dashboard, Contacts, Galleries, Hostels, MealMenu कन्ट्रोलरहरू प्रयोग गर्न सक्छ

2. \*\*होस्टल मालिक/प्रबन्धक (Owner)\*\*:

- आफ्नो होस्टल, कोठा, खाना, भुक्तानी, समीक्षा आदि व्यवस्थापन गर्न सक्छ

- आफ्नो संगठन/होस्टलमा मात्र सीमित

- Dashboard, Contacts, Galleries, Hostel, MealMenus, Payments, Reviews, Rooms, Students कन्ट्रोलरहरू प्रयोग गर्न सक्छ

3. \*\*छात्र (Student)\*\*:

- आफ्नो प्रोफाइल, बुकिङ, भुक्तानी, समीक्षा व्यवस्थापन गर्न सक्छ

- Admin र Owner भन्दा सीमित पहुँच

- Dashboard, Profile, Room, Search, Student कन्ट्रोलरहरू प्रयोग गर्न सक्छ

## अर्को कदम सिफारिसहरू

1. \*\*कन्ट्रोलर संरचना मानकीकरण\*\*:

- Singular वा Plural कन्ट्रोलर नामहरू एकरूपतामा प्रयोग गर्नुहोस् (RoomController vs RoomsController)

- app/Http/Controllers/Admin, app/Http/Controllers/Owner, app/Http/Controllers/Student मा कन्ट्रोलरहरू व्यवस्थित गर्नुहोस्

2. \*\*मिडलवेयर कार्यान्वयन पूरा गर्नुहोस्\*\*:

- सबै भूमिका-विशिष्ट कन्ट्रोलरहरूमा उपयुक्त मिडलवेयर लागू गर्नुहोस्

- CheckRole.php र RoleMiddleware.php को प्रभावकारिता परीक्षण गर्नुहोस्

3. \*\*भ्यु संरचना पूरा गर्नुहोस्\*\*:

- resources/views/admin, resources/views/owner, resources/views/student अनुसार भ्युहरू व्यवस्थित गर्नुहोस्

- सबै कन्ट्रोलर कार्यहरूका लागि संगत भ्युहरू सुनिश्चित गर्नुहोस्

4. \*\*API अन्तिम बिन्दुहरू पूरा गर्नुहोस्\*\*:

- आवश्यक भएका कन्ट्रोलरहरूका लागि API संस्करणहरू लागू गर्नुहोस्

- API अन्तिम बिन्दुहरूका लागि उचित अनुरोध प्रमाणीकरण सुनिश्चित गर्नुहोस्

5. \*\*फर्म रिक्वेस्ट प्रमाणीकरण पूरा गर्नुहोस्\*\*:

- सबै फर्म रिक्वेस्ट क्लासहरूमा प्रमाणीकरण नियमहरू थप्नुहोस्

6. \*\*मोडेलहरूलाई सम्बन्ध सहित पूरा गर्नुहोस्\*\*:

- सबै मोडेलहरूमा Eloquent सम्बन्धहरू परिभाषित गर्नुहोस्

- डाटाबेस माइग्रेसनहरूले मोडेल सम्बन्धहरूलाई सही ढंगले प्रतिनिधित्व गर्दछन् भन्ने सुनिश्चित गर्नुहोस्

7. \*\*ड्यासबोर्ड कार्यक्षमता पूरा गर्नुहोस्\*\*:

- प्रत्येक भूमिकाका लागि ड्यासबोर्डले प्रासंगिक जानकारी देखाउन सक्ने बनाउनुहोस्

- विश्लेषण र प्रतिवेदन सुविधाहरू थप्नुहोस्