

Laravel Link

It is an online platform that is dedicated to connecting Laravel developers to web development clients. This means the whole point of the website is to enable laravel developers to collaborate with clients toward the execution of software projects made in laravel. The main goal of the platform is to enable clients to get their projects done and freelancers to earn a living off their skills.

For a client to achieve this goal, they have to follow the following general steps:

- Sign up on the platform
- Post a project
- Receive proposals from freelancers
- Vet proposals
- Interview freelancer
- Agree with freelancer on a set of project terms
- Sign a contract
- Deposit project cash into escrow
- Hire the freelancer
- Upon project completion, the freelancer submits the project to client
- Client approves the project
- Project money is then released from escrow to freelancer's account

Based on these simple steps, the following list showing a high level description of the interactions between LaravelLink's users and the system can be deduced:

1. Clients

- Manage Profile
- Oversee Projects
- Manage Wallet
- Hire and Collaborate with Freelancers
- Request Support from Admins / Resolve Issues with Admins
- Network with Other Clients

2. Freelancers

- Manage Profile
- Manage Wallet
- Submit and Track Proposals
- Work on Assigned Projects
- Network with Other Freelancers.
- Seek Support from Admins
- Communicate and Deliver Work to Clients.

3. Admins

- Oversee and Support Clients
- Oversee and Support Freelancers
- Review and Moderate Proposals.
- Manage Wallets
- Monitor and Resolve Project Issues.
- Provide Support to Clients.
- Assist Freelancers with Platform Use.

Chapter 1: Core Features

The core feature of Laravel Link should cater for three kinds of users mentioned earlier. This is why the core features section is going to be divided into three sections.

Clients

The following is a list of core features that are important to a client:

1.1 Account Creation

1.1.1 Sign Up

Workflow

- Load sign up page
 - Display the sign-up form with the required fields for the client to fill out
- Fill out form and submit
 - User fills out form and input is validated according to the input field and validations rules
 - If validation fails, errors are displayed next to the affected fields
 - On successful sign up, redirect the user to the email confirmation page with a success message
- Verify Email Address
 - An email is sent to the provided email address with a confirmation link
 - The link expires after a specified time
 - If the link is valid and clicked, the user is redirected to their dashboard

Input Fields and Validation

- First Name: [required, string, min:1_char, max:50_chars]
- Surname: [required, string, min:1_char, max:50_chars]
- Email: [required, email, max:255_chars, unique]
- Password: [required, string, min:8, confirmed, Password::defaults()]
- Profile Picture: [required, image, mimes:jpeg,png,jpg,gif, max:2kb, square, minWidth:300, maxWidth:300]
- Location
 - Country: [string, required, in:[list of country codes]]
 - City: [string, required, in:[valid list of cities for selected country]]
- Website Link: [optional, url]
- Bio: [required, string, min:150, max:1000, Basic character validation, without HTML tags]
- Social Media Links: [optional]

1.1.2 Login

Workflow

- Load login page
 - Display the login form with email and password fields
 - Provide options for 'Forgot Password' and a link to the sign up page for new users

- User fills out and submits form
 - Inputs are validated against specified rules
 - If validation fails, display errors next to the affected fields
 - If validation succeeds, check the credentials against the database
- System actions based on credential validation
 - If credentials match
 - Log the user in and redirect them to their dashboard
 - If credentials don't match
 - Display an error message

Input Fields and Validation

- Email: [required, email, matches stored email]
- Password: [required, string, matches stored password]

1.2 Project Management

1.2.1 Create Project

Input Fields and Validation

- Title: [required, string, min:5, max:100, without HTML tags]
- Categories: [required, in: [list of predefined categories]]
- Description: [required, string, min:50, max:2000, without HTML tags]
- Budget: [required, integer, min:\$10, max:\$5000, in:[List of valid currencies]]
- Duration: [required, allowed units: days (max:6), weeks (max:4), months (positive number)]
- Difficulty Level: [required, in:[easy, intermediate, hard]]
- Language: [required, in:[Predefined list of languages]]
- Skills: [required, in: [predefined list of skills]]
- Attachments: [optional, file, max size: 5MB, allowed formats: PDF, DOCX, PNG, JPG]

Workflow

- Load Client Dashboard
 - Display the client dashboard
 - Highlight the "Create Project" button in a prominent location.
- Initiate Project Creation:
 - The client clicks the "Create Project" button to load the project creation form.
- Fill Out the Form and Submit:
 - The client completes the input fields based on the required criteria.
 - Upon clicking "Publish Project":
 - Validation: The system checks all fields against the defined validation rules.
 - If validation fails:
 - Display error messages directly next to the affected fields.
 - If validation succeeds:
 - Save the project details to the database.
 - Post-Creation Actions:

- Redirect the client to their "Open Projects" listing with a success message

1.2.2 View Projects

This feature allows clients to view a list of all their projects with options to filter, sort and view a project in detail.

Workflow

- Load Projects Listing
 - Display a paginated list of the client's projects on the dashboard
 - Each project entry should include the following key details:
 - Project Title
 - Category
 - Elapsed Time (e.g., "Posted 3 days ago")
 - Number of Proposals Sent
 - Budget
 - Project Status (e.g., "Open", "Active", "Completed")
 - Display pagination controls for easy navigation
- Search, Filter and Sort Projects
 - Search Projects
 - A search is provided for the client to search for specific projects by title or keyword
 - Filter Projects
 - Clients can filter projects based on the following criteria:
 - Category
 - Location
 - Price (Within a specified range)
 - Language
 - Skills Required
 - Project Title
 - Project Status
 - Filter should be easily accessible and allow multiple filters to be added simultaneously
 - Sort Projects
 - Clients can sort the project list by:
 - Date Posted (ascending/descending)
 - Budget (ascending/descending)
 - Number of Proposals Sent (ascending/descending)
 - Status (optional: Sort by project status)
 - Filter and sort options should be user-friendly and intuitive. A reset button should be available to clear selected filters/sorting.
- View or Manage Specific Projects:
 - Clients can click on any project from the list to view its detailed page.
 - Project Details Page:

- Display complete information about the project:
 - Title
 - Description
 - Category
 - Budget
 - Proposals
 - Due date
 - Skills
 - Status
 - Assigned freelancer
 - Attachments
 - Chat section
- Quick Actions:
 - Edit: Allow clients to edit details of their project if needed (e.g., changing project description, budget, or other details).
 - Delete: Provide a delete button for clients to remove a project from the list (with confirmation to prevent accidental deletion).
 - View Proposals: Show a list of proposals sent to the project with options to review, accept, or reject proposals.

Input Fields and Validation

- Search Bar
 - Keyword: [optional, string, max:255, without HTML tags]
- Filter Options
 - Category: [optional, in:[list of predefined categories]]
 - Location: [optional, in:[list of predefined locations]]
 - Price Range:
 - Min Price: [optional, numeric, min:1]
 - Max Price: [optional, numeric, greater_than:min_price]
 - Language: [optional, in:[list of predefined languages]]
 - Skills Required: [optional, in:[list of predefined skills]]
 - Project Title: [optional, string, max:100, without HTML tags]
 - Project Status: [optional, in:[open, active, in_revision, completed]]
- Sort Options
 - Sort By: [required, in:[date_posted, budget, num of proposals, status]]
 - Order: [required, in:[asc, desc]]

- Chat Message: [required, string, min:1, max:2000, without HTML tags]

1.2.3 Edit Projects

This feature enables clients to modify the details of their open projects while maintaining validation integrity and user experience.

Workflow

1. Load Project Listing:
 1. Display the client's open projects in a paginated list.
 2. Each project entry includes an "Edit" button for accessible modification.
2. Choose an Open Project:
 1. Only projects with the status Open can be edited.
 2. Clicking "Edit" loads the project details in an editable form.
3. Edit and Publish Project:
 1. Client modifies any field in the form.
 2. Validation checks ensure all input fields meet the defined criteria.
 3. If validation passes:
 - Save the updated project details to the database.
 - Redirect the user to their updated Open Projects listing.
 4. If validation fails:
 - Highlight errors next to the invalid fields.

Input Fields and Validation

- Title: [required, string, min:5, max:100, without HTML tags]
- Categories: [required, in:[list of predefined categories]]
- Description: [required, string, min:50, max:2000, without HTML tags]
- Budget: [required, integer, min:10, max:5000, in:[list of valid currencies]]
- Duration: [required]
 - Days: [integer, max:6]
 - Weeks: [integer, max:4]
 - Months: [integer, positive]

- Difficulty Level: [required, in:[easy, intermediate, hard]]
- Language: [required, in:[list of predefined languages]]
- Skills: [required, in:[list of predefined skills]]
- Attachments: [optional, file, mimes:jpeg,png,jpg,gif,pdf,doc,docx, max:5MB each, max:3 files]

1.2.4 Delete Projects

Workflow

1. Load Project Listing

- Display a list of the client's projects.
- Include a Delete button for each project in the list or in the project details page.
- Only Open projects should have a delete option.

2. Select a Project to Delete

- The client clicks the Delete button for a specific project.
- The system displays a confirmation modal or popup with the message:

"Are you sure you want to delete this project? This action cannot be undone."

3. Confirm Deletion

- If the client confirms the deletion:
 - Delete project in the database.
 - Redirect the client to the Project Listing page with a success message:

"Project deleted successfully."

- If the client cancels the action, close the modal and return to the current page.

4. Handle Errors (Optional)

- If the deletion fails due to a system error or network issue, display a clear error message:

"An error occurred while attempting to delete the project. Please try again."

1.2.5 Manage Proposals

Workflow

1. Load Project Details

- Client navigates to the Project Listing page and selects a specific project.
- Display complete details of the selected project, including a tab or section for proposals.

2. Load Proposals Listing

- Display a paginated list of proposals submitted for the selected project.
- For each proposal, show key details:
 - Freelancer's name
 - Proposal summary
 - Budget
 - Proposed Duration
 - Skills matching the project requirements
- Include filters and sorting options to help the client organize proposals (e.g., by budget, skills, or freelancer rating).

3. Select and Load a Specific Proposal

- Client clicks on a proposal to view its full details, including:
 - Freelancer's complete proposal description
 - Detailed timeline and milestones
 - Quoted budget and payment terms
 - Freelancer's profile, portfolio, and rating/reviews

4. Evaluate Proposal

- By default, submitted proposals have a pending status.
- The client has the following options for each proposal:
 - Shortlist Proposal:
 - Sends a custom message to the freelancer to schedule an interview.
 - Updates the proposal status to Shortlisted.
 - Reject Proposal:
 - Sends a default rejection message to the freelancer.
 - Updates the proposal status to Rejected.

5. Communicate with Freelancer

- After accepting a proposal, the client can chat with the freelancer via the LaraveLLink chat system to discuss further details.
- Display a chat interface for real-time communication.
- If necessary, the client can negotiate terms or clarify project details before scheduling an interview.

6. Schedule Interview

- Upon agreeing on an interview time via chat, the client:
 - Schedules a Zoom meeting with the freelancer.
 - Posts the meeting details (time, date, and link) in the chat system.
- The system notifies the freelancer of the scheduled interview via email and system notifications.

7. Post-Decision Workflow

- Accepted Proposals:
 - Transition the project to the next stage (e.g., assign the project to the selected freelancer).
- Rejected Proposals:
 - The proposal is archived in the freelancer's profile under "Rejected Proposals."
 - Notify the freelancer with a system message about the rejection.

Input Fields and Validation

1. Proposal Filters and Sorting

- Budget Range: [min:integer, max:integer]
- Skills: [in:[list of predefined skills]]
- Status: [in:[Accepted, Shortlisted, Rejected, Pending]]
- Sort By: [in:[Budget (asc/desc), Freelancer Rating, Submission Date]]

2. Proposal Evaluation

- Proposal ID: [required, exists:proposals,id]
- Action: [required, in:[Accept, Shortlist, Reject]]
- Custom Message (optional for Accept):
 - [string, min:10, max:500]
- Custom/Default Message (for Shortlist):
 - [string, min:10, max:500]
- Default Message (auto-generated for Reject):
 - [string, predefined template based on rejection reason]

3. Chat Communication

- Message Content: [required, string, min:1, max:2000, without HTML tags]
- Attachments (optional): [file, max:5MB, allowed_types:jpeg,png,doc,pdf]

1.2.6 Sign Contract

- Work Flow
 - Freelancer Prepares Contract
 - After an interview, the freelancer drafts a contract within LaraveLLink using a contract creation form. The following details are provided:
 - Project Title
 - Payment Terms (All projects have fixed price).
 - Amount
 - Timeline and Deadlines
 - Deliverables
 - Save and Send Contract
 - Once the contract is complete, the freelancer reviews and submits it.
 - The contract is automatically sent to the client via LaraveLLink's chat system and to their email as well
 - The client receives a notification about the new contract.
 - Client reviews contract
 - The client opens the contract for review
 - Options:
 - Sign: If satisfied, the client deposits funds into escrow and signs the contract
 - Request revisions: The client suggest changes via LaraveLLink's chat system
 - Client signs contract
 - When signing, the client:
 - Enters their full name.
 - Checks a box to confirm agreement with the terms.
 - After signing, both parties receive a copy of the finalized contract
- Key Features
 - Contract Creation Form
 - Input Fields and Validation
 - Project Title: [string, min:5, max:100, without html tags, from project title]
 - Payment Terms (All projects have fixed price). [Fixed]
 - Amount: [integer, min:\$10, max:\$5000, in:[List of valid currencies]]
 - Deadline: [date, should be in the future]
 - Deliverables: [string, min:50, max:2000]
 - Contract Management
 - Both parties can view contract status (Pending signature, Signed, Revised, Need revision)
 - Notifications and Emails
 - Clients are notified when a contract is ready for review.

- Freelancers are notified when the client signs or requests changes.
- Actions
 - Freelancer Actions:
 - Draft and save a contract.
 - Send the contract to the client.
 - Edit and resend contracts if revisions are requested.
 - Client Actions:
 - Review the contract.
 - Sign the contract or request revisions.
 - Platform Actions:
 - Store contracts securely.
 - Track the status of contracts.
 - Notify parties about contract updates.

1.2.8 Monitor Project Progress

The feature enables clients to communicate with freelancers via the chat system.

- Work Flow
 - Client loads active projects
 - Client selects project and communicates with freelancer via project chat

1.2.9 Archive Completed Projects

- Work Flow
 - Freelancer completes project
 - The freelancer finalizes the deliverables and submits a review request to the client via the platform.
 - Client reviews the project via a zoom meeting
 - A scheduled Zoom meeting allows the client and freelancer to discuss the project deliverables in detail.
 - During the meeting, the client evaluates the project and chooses one of the following options:
 - Requests revisions
 - Action:
 - The client uses the chat system to communicate the required changes or refinements.
 - Platform Workflow:
 - A notification is sent to the freelancer, highlighting the requested revisions.
 - The project status is updated to "In Revision."
 - The freelancer works on the revisions and resubmits the updated deliverables.
 - Approves projects
 - Action:
 - The client approves the project, signaling completion.
 - Platform Workflow:
 - Funds Released:

- The client releases funds from the escrow account to the freelancer's account.
- Submission of Final Deliverables:
 - The freelancer submits the final project outcomes to the client
- Security Waiting Period:
 - To ensure client satisfaction and security, the funds remain in a pending state for a set period (e.g., 5-7 days) before becoming withdrawable by the freelancer.
- Project Marked as Complete:
 - The project is officially marked as "**Completed**" in the system.
- Client Feedback:
 - The client is prompted to leave a review and rating for the freelancer.
 - Feedback includes details on the freelancer's communication, adherence to deadlines, and quality of work.
- Freelancer Benefits:
 - A good rating boosts the freelancer's profile visibility.

Freelancers

2.1 Account Creation

2.1.1 Sign Up

Workflow

- Load sign up page
 - Display the sign-up form with the required fields for the client to fill out
- Fill out form and submit
 - User fills out form and input is validated according to the input field and validations rules
 - If validation fails, errors are displayed next to the affected fields
 - On successful sign up, redirect the user to the email confirmation page with a success message
- Verify Email Address
 - An email is sent to the provided email address with a confirmation link
 - The link expires after a specified time
 - If the link is valid and clicked, the user is redirected to their dashboard

Input Fields and Validation

- First Name: [required, string, min:1_char, max:50_chars]
- Surname: [required, string, min:1_char, max:50_chars]

- Email: [required, email, max:255_chars, unique]
- Password: [required, string, min:8 , confirmed, Password::defaults()]
- Profile Picture: [required, image, mimes:jpeg,png,jpg,gif, max:2kb, square, minWidth:300, maxWidth:300]
- Location
 - Country: [string, required, in:[list of country codes]]
 - City: [string, required, in:[valid list of cities for selected country]]
- Website Link: [optional, url]
- Bio: [required, string, min:150, max:1000, Basic character validation, without HTML tags]
- Social Media Links: [optional]

2.1.2 Login

Workflow

- Load login page
 - Display the login form with email and password fields
 - Provide options for 'Forgot Password' and a link to the sign up page for new users
- User fills out and submits form
 - Inputs are validated against specified rules
 - If validation fails, display errors next to the affected fields
 - If validation succeeds, check the credentials against the database
- System actions based on credential validation
 - If credentials match
 - Log the user in and redirect them to their dashboard
 - If credentials don't match
 - Display an error message

Input Fields and Validation

- Email: [required, email, matches stored email]
- Password: [required, string, matches stored password]

2.2 Submit Project Proposals

Input Fields and Validation

- **Title:** [required, string, min:10, max:150, without HTML tags]
- **Proposal Description:** [required, string, min:100, max:5000, without HTML tags]
- **Attachments:** [optional, file, max size: 10MB, allowed formats: PDF, DOCX, PNG, JPG]
- **Budget:** [required, numeric, min:\$10, max:\$5000, in:[List of valid currencies, including Bitcoin]]
- **Timeline:** [required, string, format: "Start Date - End Date", must not exceed the client's project duration]
- **Skills Highlight:** [optional, in: [predefined list of skills], min:1, max:5]

Workflow

1. Access the Project Proposal Form

- Navigate to the project listing.
- Click the "Submit Proposal" button next to the desired project.

2. Load Proposal Form

- Dynamically pre-fill project details (e.g., categories, expected duration) for reference.
- Display validation rules as tooltips next to each field.

3. Fill Out the Proposal and Attach Files

- Freelancer enters the proposal details following the input field rules.
- Attachments can be uploaded or linked via cloud storage integration.

4. Submit Proposal

- Click the "Submit Proposal" button to validate the input fields.
 - **If validation fails:**
 - Display error messages inline, next to the invalid fields.
 - **If validation succeeds:**
 - Save the proposal to the database and associate it with the selected project.

5. Post-Submission Actions

- Display a success message confirming the proposal submission.
- Redirect the freelancer to the "Submitted Proposals" section in their dashboard.
- Notify the client about the new proposal.

6. Track Proposal Status

- Update the status dynamically as the client reviews and responds to the proposal.
- Notify the freelancer of status changes: **Under Review**, **Accepted**, or **Rejected**.

2.3 Proposal Negotiation

Input Fields and Validation

- Message Content: [Required, string, min:10, max:2000, without HTML tags]
- Attachments: [Optional, file, max size: 10MB, allowed formats: PDF, DOCX, PNG, JPG]
- Contract Document: [Required during agreement submission, file, max size: 5MB, allowed formats: PDF]
- Zoom Link: [Optional, valid URL format]

- Agreement Terms: [Required, string, min:50, max:2000, without HTML tags]
- Digital Signature (Client): [Required, must match stored signature pattern]

Workflow

1. Notification to Freelancer:
 - Freelancer receives a notification of the client's message via LaravelLink and email.
2. Access Project Chat:
 - Freelancer logs into LaravelLink and opens the project-specific chat to initiate discussions with the client.
3. Optional Video Chat:
 - The client schedules a Zoom video chat, if necessary, to discuss the terms in more detail.
4. Agreement on Terms:
 - Once both parties agree on the terms, the freelancer drafts and submits a formal contract.
5. Contract Review and Approval:
 - The client reviews the contract and signs it digitally to finalize the terms of the project.

2.4 Work Commencement

Input Fields and Validation

- Project Start Date: [Required, valid date, cannot be earlier than the current date]
- Progress Updates: [Required at milestones, string, min:20, max:2000, without HTML tags]
- Deliverables: [Required, file, max size: 20MB, allowed formats: ZIP, PDF, PNG, JPG, DOCX]
- Revision Requests: [Optional, string, min:10, max:1000, without HTML tags]
- Escrow Release Confirmation: [Required, boolean, must match escrow balance]

Workflow

1. Client Approves Proposal:
 - The client officially accepts the proposal by signing it via LaravelLink.
2. Freelancer Begins Work:
 - The freelancer starts working on the project, following the agreed-upon scope and timeline.
3. Client Monitors Progress:
 - The client uses tools like progress tracking, milestone updates, or regular reports to monitor the project's status.
4. Freelancer Completes Work:
 - The freelancer submits the final deliverables to the client for review.
5. Client Reviews Work:
 - **Satisfied Client:** If the client is satisfied with the work, they release the funds held in escrow.
 - **Dissatisfied Client:** If revisions are needed, the client requests changes, sending the project workflow back to Step 2 for adjustments.
6. Fund Withdrawal:
 - Upon approval, the freelancer withdraws the released funds from their LaravelLink account.

7. Project Closure:

- The project is marked as complete, and both parties can leave reviews and ratings for each other.

Guests

1. Browse Projects
2. View Freelancer Profiles
3. View Client Profiles
4. Search Projects
5. Search Freelancers
6. View Platform Statistics
7. Sign Up
8. Contact Support

Browse Projects

Input Fields and Validation

- None (Browsing does not require input fields).

Workflow

1. Guest visits LaravelLink and selects the “Browse Projects” option from the homepage.
2. Guest views a list of available projects, including basic information like title, budget, duration, and required skills.
3. Guest clicks on a project to view its details, but detailed contact or application options are restricted to registered users.

View Freelancer Profiles

Input Fields and Validation

- None (Viewing does not require input fields).

Workflow

1. Guest navigates to the “Freelancers” section on the platform.
2. Guest views a list of freelancer profiles with limited public information, such as name, skills, ratings, and completed projects.
3. Guest clicks on a profile for more details but cannot directly contact freelancers unless registered.

View Client Profiles

Input Fields and Validation

- None (Viewing does not require input fields).

Workflow

1. Guest selects the “Clients” section from the main menu.
2. Guest browses through client profiles to see basic details, such as their industry, completed projects, and average ratings.
3. Guest clicks on a profile to view more details but cannot initiate contact unless registered.

Search Projects

Input Fields and Validation

- Search Term: [optional, string, max:100]
- Category: [optional, in:[list of predefined categories]]
- Budget Range: [optional, numeric range, min:10, max:5000]
- Duration: [optional, in:[days, weeks, months]]
- Difficulty Level: [optional, in:[easy, intermediate, hard]]

Workflow

1. Guest types keywords or uses filters to search for projects.
2. System validates the input and displays matching results.
3. Guest can view a limited description of the projects but needs to register to apply or contact clients.

Search Freelancers

Input Fields and Validation

- Search Term: [optional, string, max:100]
- Skills: [optional, in:[predefined list of skills]]
- Location: [optional, valid country or city name]
- Rating: [optional, numeric range, min:1, max:5]

Workflow

1. Guest types keywords or uses filters to search for freelancers.
2. System validates the input and displays matching results.
3. Guest views limited details of freelancer profiles but cannot contact them unless registered.

View Platform Statistics

Input Fields and Validation

- None (Viewing does not require input fields).

Workflow

1. Guest visits the homepage or a dedicated “Statistics” page.
2. Guest views metrics like total projects posted, active freelancers, and completed projects.
3. No interaction is required.

Sign Up

Input Fields and Validation

- Full Name: [required, string, min:3, max:50]
- Email Address: [required, valid email format, unique]
- Password: [required, string, min:8, max:50, contains:[uppercase, lowercase, number, special character]]
- User Type: [required, in:[Freelancer, Client]]
- Agreement to Terms: [required, boolean]

Workflow

1. Guest clicks the “Sign Up” button on the homepage.
2. Guest completes the registration form with valid inputs.
3. System validates inputs and creates an account upon successful registration.
4. Guest is redirected to their dashboard after signing up.

Contact Support

Input Fields and Validation

- Full Name: [required, string, min:3, max:50]
- Email Address: [required, valid email format]
- Message Content: [required, string, min:10, max:1000, without HTML tags]
- Attachment: [optional, file, max size: 5MB, allowed formats: PDF, PNG, JPG]

Workflow

1. Guest navigates to the “Contact Us” page and fills out the support form.
2. System validates the inputs.
3. Upon successful validation, the support request is sent to the LaravelLink support team.
4. Guest receives a confirmation email with a ticket ID for tracking.

Features for Admin

1. Manage Users
2. Moderate Projects and Proposals
3. Handle Dispute Resolutions
4. Manage Payments and Escrow
5. Monitor Platform Statistics
6. Send Notifications and Announcements
7. Approve/Reject Freelancer Verification Requests
8. Manage Categories and Skills
9. Manage Reports and Feedback
10. Manage Support Tickets

Manage Users

Input Fields and Validation

- Search Term: [optional, string, max:100]

- User Type: [optional, in:[Freelancer, Client]]
- Status: [optional, in:[Active, Suspended]]

Workflow

1. Admin logs into the dashboard and selects “Manage Users.”
2. Admin searches for users or filters them by type or status.
3. Admin views user profiles, including activity history and account details.
4. Admin can suspend, delete, or restore user accounts with justifications.

Moderate Projects and Proposals

Input Fields and Validation

- Search Term: [optional, string, max:100]
- Category: [optional, in:[list of predefined categories]]
- Status: [optional, in:[Pending, Approved, Rejected]]

Workflow

1. Admin accesses the “Moderate Projects and Proposals” section.
2. Admin reviews flagged or newly submitted projects and proposals.
3. Admin approves, rejects, or requests edits, providing feedback to the submitter.
4. Decisions are communicated to the concerned users via notifications.

Handle Dispute Resolutions

Input Fields and Validation

- Ticket ID: [required, valid ticket number format]
- Dispute Summary: [required, string, min:50, max:1000, without HTML tags]
- Resolution Comments: [optional, string, max:1000]

Workflow

1. Admin navigates to “Dispute Resolutions” and selects an active ticket.
2. Admin reviews evidence and comments from both parties.
3. Admin consults terms and conditions to reach a resolution.
4. Admin updates the ticket with the decision and closes it, notifying all parties.

Manage Payments and Escrow

Input Fields and Validation

- Search Term: [optional, string, max:100]
- Transaction ID: [optional, valid format]
- User Type: [optional, in:[Freelancer, Client]]
- Status: [optional, in:[Pending, Completed, Failed]]

Workflow

1. Admin selects “Payments and Escrow” from the dashboard.

2. Admin searches or filters transactions based on criteria.
3. Admin reviews payment details, including amounts, dates, and recipients.
4. Admin resolves payment disputes, processes refunds, or releases escrow funds.

Monitor Platform Statistics

Input Fields and Validation

- None (Viewing does not require input fields).

Workflow

1. Admin selects “Platform Statistics” from the dashboard.
2. Admin views metrics like active users, completed projects, and transaction volumes.
3. Admin downloads reports or shares insights with the team if needed.

Send Notifications and Announcements

Input Fields and Validation

- Title: [required, string, min:5, max:100, without HTML tags]
- Message Content: [required, string, min:10, max:2000, without HTML tags]
- Target Audience: [required, in:[All Users, Freelancers, Clients]]

Workflow

1. Admin navigates to “Notifications and Announcements.”
2. Admin composes a message with a title and content.
3. Admin selects the target audience and schedules or sends the message immediately.
4. The notification appears on users’ dashboards and is sent via email.

Approve/Reject Freelancer Verification Requests

Input Fields and Validation

- Freelancer ID: [required, valid user ID]
- Verification Document: [required, file, max size: 5MB, allowed formats: PDF, PNG, JPG]
- Admin Comments: [optional, string, max:500]

Workflow

1. Admin accesses “Freelancer Verification Requests.”
2. Admin reviews submitted documents for authenticity.
3. Admin approves or rejects the request, providing comments if needed.
4. The freelancer is notified of the decision.

Manage Categories and Skills

Input Fields and Validation

- Category Name: [required, string, min:3, max:50, unique]
- Skill Name: [required, string, min:3, max:50, unique]

Workflow

1. Admin selects “Manage Categories and Skills” from the dashboard.
2. Admin adds, edits, or deletes categories and skills.
3. Changes are instantly reflected across the platform.

Manage Reports and Feedback

Input Fields and Validation

- Report ID: [required, valid report number format]
- User ID: [required, valid user ID]
- Resolution Comments: [optional, string, max:1000]

Workflow

1. Admin accesses “Reports and Feedback.”
2. Admin reviews flagged content or user complaints.
3. Admin resolves reports by taking necessary actions, such as warnings, suspensions, or deletions.
4. Admin logs the resolution and notifies the reporter.

Manage Support Tickets

Input Fields and Validation

- Ticket ID: [required, valid ticket number format]
- User ID: [required, valid user ID]
- Response Message: [required, string, min:10, max:2000]

Workflow

1. Admin navigates to “Support Tickets.”
2. Admin selects an unresolved ticket and reviews its details.
3. Admin responds to the ticket or escalates it to the technical team.
4. The ticket is marked as resolved, and the user is notified.

Laravel Link Predefined Lists

Project Categories

1. Web Application Development
2. API Development
3. Package Development
4. E-commerce Solutions
5. CMS Development
6. Mobile App Backend
7. Frontend Development
8. Database Optimization
9. Admin Dashboards
10. Authentication Systems
11. Bug Fixes & Maintenance
12. Security Enhancements
13. Performance Tuning
14. Version Upgrades
15. Automated Testing
16. Deployment & Hosting
17. Third-Party Integrations
18. Consultancy & Architecture
19. Project Audits
20. SaaS Development
21. Real-Time Apps
22. Social Media Integration
23. Analytics Dashboards
24. SEO Optimization
25. Localization & l18n
26. Enterprise Solutions
- 27. Freelancer/Client Portals**

Project Statuses

1. Open
2. Active
3. In Revision
4. Completed

Freelancer Skills

1. MVC
2. Blade
3. HTML, CSS
4. JavaScript
5. Tailwind CSS
6. Bootstrap
7. Vue.js
8. React
9. Alpine.js
10. MySQL
11. PostgreSQL
12. SQLite
13. Laravel Migrations
14. Laravel Seeds
15. Laravel Factories
16. Eloquent ORM
17. RESTful API Development
18. Laravel Passport
19. JWT Authentication
20. API Rate Limiting
21. Swagger
22. PHPUnit Testing
23. Laravel Dusk
24. Xdebug
25. Laravel Debugbar
26. Git

27. GitHub
28. GitLab
29. Bitbucket
30. AWS
31. DigitalOcean
32. Heroku
33. Vultr
34. Nginx
35. Apache Configuration
36. CI/CD Pipelines
37. Laravel Envoyer
38. Laravel Queues
39. Laravel Scheduler
40. CSRF Protection
41. XSS Protection
42. SQL Injection Protection
43. Laravel Encryption
44. Two-Factor Authentication (2FA)
45. Redis
46. Memcached
47. Query Optimization
48. Caching Strategies
49. Docker
50. Linux Server Management
51. SSH
52. React Native
53. Flutter
54. SEO Optimization
55. Image Optimization
56. Responsive Design

Freelancers

1. Submit Project Proposals
2. Set Custom Pricing Packages
3. View Earnings in Multiple Currencies (including Bitcoin)
4. Withdraw Funds in Bitcoin or Local Currency
5. Access Proposal Templates
6. Track Proposal Status (Pending, Accepted, Rejected)
7. Manage Subscription Plans for Platform Features
8. Generate Payment Requests for Milestones

Clients

1. Post Detailed Project Requirements
2. View and Compare Freelancer Proposals
3. Pay in Bitcoin or Local Currency
4. Set Milestone Payments with Release Triggers
5. Approve or Reject Proposal Revisions
6. Request Freelancer Availability for Future Projects
7. Access Payment History for Accounting

Admins

1. Monitor Bitcoin Payment Gateway Transactions
2. Resolve Proposal Conflicts
3. Oversee Subscription Management for Freelancers
4. Audit Proposal Quality and Reject Spam Submissions
5. Generate Escrow Payment Reports

6. Configure Platform-Wide Bitcoin Exchange Rates

7. Approve Client and Freelancer Payment Requests