### **System Flow:**

1. **User Submits a Request:**
   * **Form Fields:**
     + Project Name
     + Project Description
     + Task Points
     + Figma Link (optional)
     + Screenshot (optional)
     + Video (optional)

**Example Request:**vbnet  
Copy code  
Project Name: LinkedIn Post Creation

Description: Create a professional LinkedIn post announcing our new product launch.

Task Points:

1. Write engaging copy for the post (100 words).

2. Design a banner image using the product logo.

Figma Link: (optional link)

Screenshot: (optional file upload)

Video: (optional file upload)

1. **Admin Receives the Request and Responds:**
   * **Admin Views:** The admin sees the LinkedIn post creation request and its details.
   * **Response Fields:**
     + Proposed Price (e.g., $100 for writing the post and designing the banner)
     + Proposed Time (e.g., 2 days to complete)
     + Additional Comments (e.g., “The post will include up to 3 design revisions.”)

**Example Admin Response:**yaml  
Copy code  
Proposed Price: $100

Proposed Time: 2 days

Comments: The post will include up to 3 design revisions and a professional banner.

1. **User Reviews Admin’s Response:**
   * **User Options:**
     + Approve
     + Request Change (e.g., “Can it be done for $75 in 1 day?”)
     + Deny

**Example User Response:**bash  
Copy code  
Request Change:

- Can we do it for $75 in 1 day instead?

1. **Admin Adjusts the Proposal (if needed):**
   * **Admin Response:** The admin can accept the counterproposal or provide another alternative.

**Example Admin Response to Change Request:**ruby  
Copy code  
Revised Proposal: $75, 1 day

Comments: This will include a single revision instead of 3.

1. **User Approves the Task:**
   * **Approval:** Once the user is satisfied with the terms, they approve the task.
   * **Both parties confirm:** The task is now confirmed with both sides.

**Example Confirmation:**yaml  
Copy code  
Project Name: LinkedIn Post Creation

Approved Price: $75

Approved Time: 1 day

1. **Thread (Comments) for Task Discussion:**
   * Once approved, a **comments section** is opened for further communication. This serves as a thread where both the user and the admin can discuss details (e.g., sharing design ideas, clarifying task points).

**Example Discussion in Thread:**vbnet  
Copy code  
User: Can you use our blue and white brand colors for the banner?

Admin: Sure! I’ll incorporate those colors into the design.

1. **Task Completion and Thread Closure:**
   * Once the admin completes the task (e.g., submits the LinkedIn post copy and banner design), both sides confirm that the task is done.

**Example Completion:**sql  
Copy code  
Admin: The LinkedIn post copy and banner design are complete. Please review.

User: I’ve reviewed, and everything looks great. Task complete!

### **Backend and Frontend Fields:**

1. **Backend:**
   * Project Management (Project ID, Name, Description, Task Points)
   * Admin Response (Price, Time, Comments)
   * User Response (Approve, Request Change, Deny)
   * Thread/Comments (Message ID, User/Admin messages, timestamps)
   * Confirmation Status (Admin/User Confirmation, Task Complete)
2. **Frontend:**
   * Request Submission Form
   * Admin Response Form
   * User Response (Approve/Change/Deny buttons)
   * Thread/Comments Section (Display messages between user and admin)
   * Task Completion and Confirmation buttons

This flow clearly outlines how users and admins interact in an on-demand resource request system, with an example task such as creating a LinkedIn post for $100.

==========================================

Here’s a structured breakdown of the fields for both the frontend and backend, ensuring that the user and admin can negotiate the task's time and amount, and both parties need to approve it.

### **Backend Fields:**

#### **User Management:**

1. **User ID** (unique identifier)
2. **Name** (text)
3. **Email** (text)
4. **Role** (enum: Admin, User)

#### **Project Request:**

1. **Project ID** (unique identifier)
2. **Project Name** (text)
3. **Project Description** (text area)
4. **Task Points** (array of tasks, each with task ID and description)
5. **Figma Link** (optional URL)
6. **Screenshot** (optional image file upload or URL)
7. **Video** (optional video file upload or URL)
8. **Request Date** (timestamp)
9. **Request Status** (enum: Pending, Admin Responded, User Counter-Proposed, Approved, Denied, In Progress, Completed)
10. **Negotiation Status** (enum: Negotiation Open, Negotiation Closed)
11. **User Comments** (text, optional)

#### **Admin Response:**

1. **Response ID** (unique identifier)
2. **Project ID** (reference to project)
3. **Admin Proposed Price** (number, e.g., 100 for $100)
4. **Admin Proposed Time** (integer, e.g., number of days)
5. **Admin Comments** (text, optional)
6. **Response Date** (timestamp)

#### **User Feedback:**

1. **Feedback ID** (unique identifier)
2. **Project ID** (reference to project)
3. **User Counter-Proposed Price** (number, e.g., 75 for $75)
4. **User Counter-Proposed Time** (integer, e.g., number of days)
5. **User Comments** (text, optional)
6. **User Response Date** (timestamp)
7. **Approval Status** (enum: Pending, Approved, Denied)

#### **Negotiation Fields:**

1. **Negotiation ID** (unique identifier)
2. **Project ID** (reference to project)
3. **Negotiation History** (array of proposals/counter-proposals)
   * Each entry includes: proposed price, proposed time, proposer (user/admin), date
4. **Negotiation Approval Status:**
   * **Admin Approval** (boolean, default = false)
   * **User Approval** (boolean, default = false)
   * Both parties need to approve before proceeding.

#### **Thread Management (Comments Section):**

1. **Thread ID** (unique identifier)
2. **Project ID** (reference to project)
3. **Messages** (array of comments, each with sender, message content, and timestamp)
4. **Thread Status** (enum: Open, Closed)

#### **Confirmation:**

1. **Project ID** (reference to project)
2. **Admin Confirmation** (boolean, default = false)
3. **User Confirmation** (boolean, default = false)
4. **Completion Date** (timestamp)

### **Frontend Fields:**

#### **Request Submission Form (User-facing):**

1. **Project Name** (input field, text)
2. **Project Description** (text area)
3. **Task Points** (dynamic input fields to add multiple tasks)
4. **Figma Link** (input field, URL)
5. **Screenshot** (file upload or URL input)
6. **Video** (file upload or URL input)
7. **Submit Button**

#### **Admin Response Form (Admin-facing):**

1. **Project Details** (display fields for project name, description, task points, etc.)
2. **Proposed Price** (input field, number)
3. **Proposed Time** (input field, number, days)
4. **Comments** (optional text area)
5. **Submit Button**

#### **Negotiation (User and Admin):**

1. **Price Proposal/Counter-Proposal** (input field for price)
2. **Time Proposal/Counter-Proposal** (input field for number of days)
3. **Comments** (optional)
4. **Approve Button** (if the party agrees to the proposal)
5. **Request Change Button** (to counter-propose price or time)
6. **Deny Button** (to reject the project)
7. **Approval Status Display:** Shows whether both admin and user have approved the terms.

#### **Thread (Comments Section for Discussions):**

1. **Thread Display** (shows previous messages)
2. **New Comment Input** (text area for comments)
3. **Post Comment Button**

#### **Approval Screens:**

1. **Display Project Details** (shows name, task points, price, and time)
2. **Approval Options:**
   * **Approve Button** (final confirmation from both sides)
   * **Request Change Button** (to reopen the negotiation)
   * **Deny Button** (to reject the project)

#### **Completion Confirmation:**

1. **Confirmation Display:** Shows when the task is completed.
2. **Close Thread Button** (appears when both user and admin confirm completion)

### **Negotiation Process:**

1. **User Submits Request:**
   * The user fills in the required fields (e.g., "Create LinkedIn Post" with task points like writing the post and designing the banner).
2. **Admin Reviews and Responds:**
   * The admin proposes a price and timeline for the task (e.g., $100 in 3 days).
3. **User Counter-Proposal:**
   * The user can approve, deny, or counter-propose (e.g., asking for $75 in 2 days).
4. **Admin Reviews Counter-Proposal:**
   * If the admin agrees to the new terms, they approve. If not, they can propose a new price or timeline.
5. **Both Parties Approve:**
   * Once both the user and admin agree on the terms (price and time), both click "Approve" to lock in the deal.
6. **Thread Opens for Discussion:**
   * A comment thread opens for further task-related discussions. Both parties can use this space to clarify details, share files, and finalize the work.
7. **Task Completion and Closure:**
   * When the task is completed, both sides confirm, and the thread is closed.

This structured breakdown clearly outlines all fields and processes needed for a smooth negotiation and project management system. Let me know if you need any further adjustments!