RoadSyster

Work & Schedule Organization

Project Structuring

When a freelancer starts a new project, they need a way to organize everything from the initial idea to final delivery in a clear and manageable way. This is where Project Structuring comes in.

1. Project Creation & Initial Setup

Define the Project Name and Description: The freelancer creates a new project by entering a name (e.g., “Website Redesign for Client X”) and a brief description outlining the project's overall purpose and objectives.

Project Type & Categories: Freelancers can categorize the project (e.g., graphic design, web development, content writing, etc.) so that it’s easily searchable and filtered later on.

2. Setting Clear Goals

Define Primary Project Goals: The freelancer sets the core objectives of the project. For example, "Create a fully responsive website with a modern design" or "Deliver 10 blog posts by the end of the month."

Breakdown goals into Sub-goals: Freelancers can break down large project goals into smaller, actionable goals. These smaller goals give clarity on what needs to be done and allow the freelancer to track progress more easily.

- **For web design:** "Create wireframes," "Develop homepage layout," "Test mobile responsiveness," etc.
- **For content creation:** "Research topics," "Write first draft," "Review and edit drafts," "Final revisions."

3. Project Scope & Deliverables

Define Project Scope: The scope refers to the work to be done and what is excluded from the project. For instance:

- **Scope:** Design a 10-page website, including homepage, contact form, and portfolio page. **Exclusions:** No content creation or SEO work included in the scope.

List Key Deliverables: The freelancer can list the final outputs or deliverables for each goal or sub-goal. These might be files, prototypes, or documents that need to be delivered at the end of the project. For example, in a website redesign project, deliverables might include:

1. A wireframe of the homepage.
2. Final HTML/CSS files.
3. User acceptance testing report.

4. Timeline & Milestones

Set Deadlines for Tasks and Goals: The freelancer defines when each task or goal should be completed. Deadlines and timelines help freelancers avoid last-minute rushes and ensure timely delivery of the project. This includes setting deadlines for:

- Milestones: Major phases or checkpoints of the project (e.g., "Complete initial design draft" or "Finish development phase").
- Smaller Tasks: Minor tasks within each goal (e.g., "Create homepage header," "Write homepage content," etc.).

Use a Calendar or Gantt Chart View: Freelancers can visualize the timeline using a calendar view or a Gantt chart. This makes it easier to:

- See upcoming deadlines.
- Allocate time for each task.
- Track dependencies (e.g., task B can't be completed until task A is done).

5. Task Dependencies & Priorities

Set Task Dependencies: Some tasks can't be started until others are finished (e.g., "Develop homepage layout" can't happen until "Create wireframe" is complete). The system allows the freelancer to mark dependencies, ensuring a smooth workflow and preventing tasks from being out of order.

Prioritize Tasks: Freelancers can assign priority levels to each task (e.g., low, medium, high). This way, they know which tasks to tackle first to keep the project on track.

6. Progress Tracking & Updates

Track Progress for Each Goal/Task: Freelancers can mark tasks as in progress, completed, or pending. This allows them to visually track progress and ensure nothing is overlooked.

Receive Automated Reminders: The system sends reminders or notifications as deadlines approach. This helps keep everything on track, even if the freelancer is juggling multiple projects.

Share Updates with Clients: Freelancers can share progress updates or milestones with clients to maintain transparency and keep the client in the loop. This can be done through automated emails or a client portal.

7. Adjustments & Flexibility

Adjust Timelines or Scope: If a freelancer needs to adjust the timeline (due to unexpected delays) or modify the scope (based on new client requests), the system allows easy updates to goals, deadlines, and deliverables. This flexibility is key for freelancers, as projects often evolve.

Budget Tracking: The system can also track the budget for each project and alert the freelancer if they are approaching or exceeding their budget for certain tasks or the entire project.

8. Finalizing the Project

Review & Quality Check: Once all tasks and goals are marked as completed, the freelancer can do a final review of the project. This ensures that all deliverables are met and that nothing has been missed.

Client Approval: The freelancer can request client approval for each deliverable or for the entire project. This ensures that the client is happy with the work and that everything is completed to their satisfaction.

Close the Project: Once everything is complete and the client is satisfied, the project is officially closed. The freelancer can archive it for future reference or generate reports on the work done.

Benefits of Project Structuring:

Clear Expectations: Both the freelancer and the client know exactly what needs to be done, by when, and how it will be delivered.

Organization & Focus: Breaking the project into smaller tasks helps the freelancer stay organized and focused.

Transparency: Clients can see real-time progress and feedback, reducing uncertainty and improving trust.

Efficiency: With deadlines, dependencies, and priorities clearly defined, the freelancer can stay on track and manage their time effectively.

Task & Subtask Management – Break work into manageable tasks, set priorities, and track progress.

Time Management – Log hours worked, estimate effort, and compare planned vs. actual time spent.

Deadline & Milestone Tracking – Set milestones, due dates, and receive reminders to stay on schedule.

Calendar Integration – View tasks, meetings, and deadlines in a unified calendar view.

Workload Overview – See current, upcoming, and completed work to avoid overbooking or burnout.

Flexible Workflows – Adapt task flows to different project types and working styles.

Task & Subtask management

The **Task & Subtask Management** feature is crucial for organizing work into manageable pieces, helping the freelancer stay on track, and ensuring clear communication with clients.

1. Task Creation

Client or Freelancer creates a Task: When starting a new project, a task is created, representing a major milestone or deliverable in the project (e.g., “Design Website Homepage” or “Write 5 Blog Posts”).

Tasks are high-level and describe a complete unit of work or a significant part of the project.

Task Attributes:

- **Title:** A descriptive title (e.g., "Design Website Homepage").
- **Description:** A detailed explanation of what needs to be done (e.g., "Create a fully responsive homepage design with HTML/CSS").
- **Deadline:** When the task needs to be completed.
- **Priority:** Assign a priority level to the task (e.g., High, Medium, Low).
- **Assigned To:** If working with a team, specify the freelancer responsible.

```json
{
  "task_id": "T001",
  "title": "Design Website Homepage",
  "description": "Create a fully responsive homepage design with HTML/CSS.",
  "deadline": "2026-02-01",
  "priority": "High",
  "assigned_to": "Freelancer_1"
}
```

2. Breaking Down into Subtasks

Subtasks Creation: Once a high-level task is created, it can be broken down into smaller, more manageable tasks (subtasks).

Subtasks represent specific actions or steps that need to be completed in order to accomplish the larger task. This helps freelancers track progress and prevents work from feeling overwhelming.

Subtask Attributes:

- **Name:** Short description of what needs to be done.
- **Description:** Detailed breakdown of the subtask.
- **Deadline:** Individual deadlines for each subtask (if applicable).
- **Priority:** Assign priority levels to subtasks for better focus.
- **Progress Tracking:** Indicates completion percentage (e.g., 50% completed, 80% done).

```json
[
 {
  "subtask_id": "S001",
  "task_id": "T001",
  "title": "Create Wireframe for Homepage",
  "description": "Design a wireframe layout for the homepage, ensuring all sections are clearly defined.",
  "deadline": "2026-01-15",
  "priority": "High",
  "progress": "50%"
 },
]
```

3. Progress Tracking & Status Updates

Tracking Task and Subtask Progress: Freelancers and clients can track the progress of tasks and subtasks in real-time.

Progress tracking can be done through: **Percentage** completion or **Status labels**

Notifications: The system sends notifications when a task or subtask's status changes (e.g., when a subtask is marked as completed, or when the deadline is approaching).


```
{
 "subtask_id": "S001",
 "progress": "50%",
 "status": "In Progress",
}
```


4. Task Prioritization & Deadlines

Prioritization: Freelancers can prioritize their tasks and subtasks by setting a priority (e.g., High, Medium, Low). This helps them focus on the most important tasks first.

The system can display high-priority tasks at the top of the list, helping the freelancer stay organized and on track.

Deadline Management: Tasks and subtasks can have specific deadlines that help manage time effectively.

Deadline reminders will be sent to freelancers and clients to avoid missed deadlines.


```
{
 "task_id": "T001",
 "priority": "High",
 "subtasks": [
  {
   "subtask_id": "S001",
   "priority": "High"
   "deadline": "2026-01-15"
  },
  {
   "subtask_id": "S002",
   "priority": "High"
   "deadline": "2026-01-25"
  }
  {
   "subtask_id": "S003",
   "priority": "Medium"
   "deadline": "2026-02-01"
  }
 ],
}
```


5. Collaboration and Comments

Collaboration: The system allows the client and freelancer to collaborate within tasks and subtasks, exchanging feedback or clarification.

Comments: The ability to leave comments on individual tasks or subtasks allows for effective communication between the freelancer and client.

For example, the client might leave feedback on a subtask ("The design needs more contrast") or ask questions about the task.


```
{
 "subtask_id": "S002",
 "comment": "Make the design more modern and vibrant? A brighter colour scheme would be better.",
}
```


6. Task Dependencies and Relationships

Task Dependencies: Some tasks may depend on the completion of others. For example, a subtask like “Review Homepage Design” depends on the completion of “Design Homepage in Figma”.

Visualizing Dependencies: The system can include a Gantt chart or task board view to visually represent task dependencies, showing the order of tasks and how they relate to one another.

Automatic Progress Updates: If a high-level task’s subtasks are completed, the system will automatically update the status of the main task to “Completed” (or another appropriate status).

7. Reporting & Analytics

Completion Reports: The system can generate reports on how many tasks and subtasks are completed, in progress, or overdue.

Time Tracking: If integrated with a time-tracking feature, freelancers can log time spent on specific tasks and subtasks, making it easier to bill clients and report progress.


```
{
 "project_id": "P001",
 "total_tasks": 5,
 "tasks_completed": 3,
 "tasks_in_progress": 1,
 "tasks_not_started": 1,
}
```


8. Task Completion & Review

Completion: Once all subtasks for a task are completed, the system can mark the main task as complete.

Client Review: After task completion, the client can review the work before final approval. This step might include minor revisions or adjustments to finalize the deliverable.

Final Feedback: After approval, the system allows for feedback from both the freelancer and client, helping improve future task management.

Client & Relationship Management

- **Client Profiles –** Store client details such as contact information, preferences, and relevant notes.

- **Communication Tracking –** Log emails, messages, meetings, and key discussions for easy reference.
- **Contract & Agreement Management –** Organize contracts, scopes of work, proposals, and revisions in one place.
- **Project–Client Linking –** Associate projects, tasks, and invoices with specific clients for clarity.
- **Collaboration History –** Maintain a record of past and ongoing work to support long-term relationships.
- **Feedback & Approval Tracking –** Capture client feedback, approvals, and change requests.
- **Client Status Overview –** View active, prospective, and completed clients at a glance.

Financial Management

Invoicing & Billing

- **Customizable Invoices –** Create professional, personalized invoices with logos, payment terms, and due dates.
- **Recurring Invoices –** Set up recurring billing for ongoing projects or retainer clients.
- **Multiple Payment Methods –** Offer various payment options (e.g., PayPal, bank transfer, credit cards) directly on invoices.
- **Invoice History & Tracking –** Keep a record of sent invoices and their payment status (paid, pending, overdue).
- **Automatic Payment Reminders –** Send automated reminders to clients for upcoming or overdue payments.
- **Invoice Templates –** Pre-made templates for different types of work (e.g., hourly, project-based, or flat rate).

Payment & Income Tracking

- **Payment Integration –** Sync with payment gateways (PayPal, Stripe, etc.) to automatically track incoming payments.
- **Real-Time Income Tracking –** Get a clear overview of income earned per project, client, and over specific time periods (e.g., weekly, monthly, yearly).
- **Multi-Currency Support –** For freelancers working globally, support for different currencies and conversion rates.
- **Payment Categorization –** Categorize payments by type (e.g., deposit, final payment, milestone) for clarity.

- **Revenue Reports –** Generate custom reports to see earnings per project, client, or month, which helps in understanding your financial trends.

Expense Tracking

- **Track Business Expenses –** Record all business-related expenses (software, tools, travel, etc.).
- **Receipt Scanning –** Use the app to scan and store receipts, making it easier to track expenditures and for tax purposes.
- **Expense Categories –** Categorize expenses (e.g., office supplies, marketing, subscriptions) to manage budgets and costs effectively.
- **Tax Deduction Tracking –** Identify potential tax-deductible expenses to make filing taxes easier and more accurate.
- **Project-Specific Expenses –** Allocate expenses to specific projects or clients to understand true project costs.

Tax & Financial Planning

- **Tax Estimates –** Estimate taxes owed based on income and expenses, helping freelancers save for tax season.
- **Tax Report Generation –** Generate simple tax reports to simplify the process of filing taxes.
- **Profit & Loss Statements –** Create P&L reports to track overall profitability and understand where your business stands financially.
- **Quarterly/Annual Income Overview –** Get a snapshot of income, expenses, and profit for the year or quarter to guide financial planning.
- **Savings Goals –** Set and track savings goals (e.g., for tax, emergency fund, retirement).

Financial Forecasting & Budgeting

- **Income Projections –** Estimate future income based on ongoing contracts and past work trends.
- **Expense Projections –** Predict upcoming expenses and adjust budgets accordingly.
- **Cash Flow Management –** Keep track of when income will come in versus when expenses are due, avoiding cash flow issues.
- **Budget Monitoring –** Set budget limits for projects, clients, or categories and get notified if you exceed them.

Reporting & Analytics

- **Financial Dashboards –** Visualize key financial metrics (e.g., income, expenses, tax estimates) in a customizable dashboard.
- **Detailed Financial Reports –** Generate monthly or yearly financial reports to track growth, profitability, and areas for improvement.

- **Client Payment History –** See which clients are consistently paying on time, and which may require more follow-up.
- **Profitability Analysis –** Identify which services, clients, or projects are most profitable to optimize future business decisions.

Encryption

1. Data Encryption on the Client Side:

Client encrypts all sensitive project data (e.g., project name, description, milestones) before sending it to the server.

Use strong encryption algorithms like AES (Advanced Encryption Standard) or RSA (for key exchange and digital signatures).

Encryption key (AES key or private RSA key) is never shared with the server; it's kept securely on the client side.

2. Data Storage on the Server:

Server receives and stores only encrypted data (ciphertext) in the database.

Server never has access to the decryption key, so it can't read or manipulate the data.

Use strong access controls and encryption for server-side storage to protect the ciphertext.

3. Secure Data Transmission (TLS/SSL):

All communication between the client and server is encrypted using TLS/SSL to protect data in transit.

Ensures confidentiality and integrity of the data while being transmitted (prevents man-in-the-middle attacks).

4. Client Requests Data (Decryption Process):

When the client requests the project data, the server sends back the encrypted data (ciphertext).

The client receives the ciphertext and decrypts it locally using its private key or symmetric encryption key.

5. Encryption Key Management:

Encryption keys (symmetric AES or RSA private key) are securely managed on the client-side.

Avoid hardcoding keys in the code; use secure key storage mechanisms (e.g., Keychain on mobile devices, Secure Enclaves, or hardware tokens).

Symmetric Key Exchange: If using AES, the client must securely exchange the symmetric encryption key with the server (via public-key encryption like RSA).

Key Rotation & Expiration:

Implement regular key rotation to ensure that encryption keys are updated periodically to prevent long-term exposure.

Implement key expiration policies for secure key management and re-encryption of data.

Secure Backup & Recovery:

Backups of encrypted data should also be securely stored.

Implement backup encryption to prevent unauthorized access, ensuring that backup copies of data remain secure.

Audit & Logging:

Access logs on the server should only record access attempts to the encrypted data (not the decrypted contents).

Track which users or clients request data, and ensure that logs are protected against tampering.

User Experience (UX):

Client-side encryption and decryption are handled transparently, ensuring that the user has a seamless experience.

The encryption and decryption process should be fast and efficient to prevent delays in data retrieval or project updates.

Key Advantages:

Only the client can decrypt the data—not the server—ensuring privacy and preventing data breaches on the server-side.

No plaintext data is ever exposed to the server, reducing the risk of unauthorized access or leaks from the server.