Я делаю сайт биржы, когда создаю задание у меня должно быть несколько этапов. Поиск автора, В работе, На гарантии, Завершен. Мне нужно знать что мне делать в этих вкладках. Какие поля или сценарий самой страницы, вобщем логика

**ChatGPT сказал:**

ChatGPT

Для реализации биржи с несколькими этапами выполнения задания (Поиск автора, В работе, На гарантии, Завершен), предлагаю следующую логику:

**Общие поля задания:**

1. **Название задания** — текст.
2. **Описание задания** — текст.
3. **Цена** — число.
4. **Исполнитель** — пользователь, который будет выполнять задание.
5. **Заказчик** — пользователь, который создает задание.
6. **Статус задания** — одно из значений (Поиск автора, В работе, На гарантии, Завершен).
7. **Дедлайн** — дата завершения работы.
8. **Дата начала работы** — дата.
9. **Комментарии/Обсуждения** — связь с таблицей комментариев.
10. **Прогресс работы** — процент выполнения или текстовый статус.

**Логика на каждом этапе:**

**1. Поиск автора:**

* **Статус**: Поиск автора.
* **Доступные действия для заказчика**:
  + Редактировать задание.
  + Просматривать заявки исполнителей.
  + Принять заявку исполнителя.
* **Доступные действия для исполнителей**:
  + Просматривать задание.
  + Подавать заявку на выполнение (включает комментарий, предлагаемый срок и цену, если она гибкая).
* **Поля**:
  + Заявки на задание (связь с таблицей заявок).

**2. В работе:**

* **Статус**: В работе.
* **Установлено, что исполнитель выбран**.
* **Доступные действия для заказчика**:
  + Просматривать прогресс работы.
  + Комментировать и обсуждать детали с исполнителем.
  + Отправить в доработку (если требуется).
* **Доступные действия для исполнителя**:
  + Загрузить файлы/отчеты по выполнению задания.
  + Отмечать прогресс выполнения задания.
  + Комментировать и обсуждать детали с заказчиком.
* **Поля**:
  + Прогресс работы (например, загрузка файлов, отметка выполнения этапов).

**3. На гарантии:**

* **Статус**: На гарантии.
* **Процесс завершен, но работа проходит гарантийный период**.
* **Доступные действия для заказчика**:
  + Оценить выполненную работу.
  + Принять или запросить исправления.
* **Доступные действия для исполнителя**:
  + Вносить правки (если заказчик запрашивает).
* **Поля**:
  + Отзывы и оценки.
  + Возможность привязки к гарантии (например, отслеживание временных рамок гарантии).

**4. Завершен:**

* **Статус**: Завершен.
* **Работа завершена и гарантии истекли**.
* **Доступные действия для заказчика**:
  + Оставить отзыв об исполнителе.
  + Закрыть задание.
* **Доступные действия для исполнителя**:
  + Просмотреть отзыв.
  + Задание закрыто для изменений.
* **Поля**:
  + История выполнения.
  + Отзывы и рейтинг.

**Сценарий для страницы задания:**

1. **Статус задания** влияет на то, какие действия доступны пользователям.
2. **Комментарии** и **обсуждения** доступны на каждом этапе после выбора исполнителя.
3. Все этапы должны быть привязаны к системе уведомлений (email/внутренняя система оповещений), чтобы уведомлять о смене статусов.

**Возможные улучшения:**

* Система **уведомлений**: уведомление исполнителя/заказчика о смене этапов.
* Введение **таймера** для каждого этапа, чтобы фиксировать, сколько времени затрачено на поиск исполнителя, выполнение и гарантийный срок.

Такая логика обеспечит структурированное выполнение задач на бирже с минимальными конфликтами между участниками.

4o

**Вы сказали:**

как расплатится с исполнителем

**ChatGPT сказал:**

ChatGPT

Чтобы реализовать функционал оплаты исполнителю на бирже, можно следовать следующей логике:

**1. Предоплата или Депонирование средств (эскроу-счет)**

* После того, как заказчик выбрал исполнителя, средства блокируются на эскроу-счете биржи.
* Средства остаются заблокированными до завершения работы или выхода задания из этапа "Гарантия".
* Это гарантирует исполнителю, что заказчик оплатит работу, а заказчику — что деньги не будут переведены до завершения задания.

**2. Сценарий оплаты при завершении задания:**

Когда задание переходит в статус "Завершен", начинается процесс выплаты.

**1. Проверка завершения работы:**

* Когда исполнитель завершает работу и передает её заказчику, задание переходит в статус "На гарантии". Заказчик имеет возможность проверить выполненную работу.
* Если заказчик подтверждает, что работа выполнена качественно, и не требует исправлений, он может нажать кнопку "Принять работу".
* Если требуется доработка, заказчик может отправить задание на доработку.

**2. Завершение гарантии и перевод средств:**

* После того как заказчик подтверждает, что работа выполнена (или по окончании гарантийного срока), задание автоматически переходит в статус "Завершен".
* После этого система автоматически разблокирует средства на эскроу-счете и переводит их на счет исполнителя.

**3. Комиссия платформы:**

* Биржа может взимать комиссию с каждой сделки. Например, при переводе средств на счет исполнителя система может удерживать определенный процент в качестве комиссии за использование платформы.
* Комиссия может взиматься как с исполнителя, так и с заказчика, в зависимости от политики платформы.

**4. Варианты оплаты:**

* **Электронные кошельки**: Поддержка систем оплаты, таких как PayPal, Яндекс.Деньги, Qiwi и т.д.
* **Банковские карты**: Платежи через Visa, MasterCard и другие.
* **Криптовалюты**: Оплата через Bitcoin, Ethereum и другие криптовалюты, если это актуально для вашего проекта.
* **Внутренний баланс**: Можно ввести внутренний баланс на платформе, где заказчик пополняет баланс, а потом использует эти средства для оплаты заданий.

**5. Возврат средств:**

* Если задание отменяется или работа не выполнена, у заказчика должна быть возможность запросить возврат средств, что приведет к разблокировке депонированных средств и возврату их на его счет.

**6. Безопасность и защита транзакций:**

* Все финансовые операции должны быть защищены с использованием SSL и других мер безопасности.
* Желательно использовать защищенные платежные шлюзы, такие как Stripe, PayPal, или национальные аналоги для обеспечения безопасности финансовых транзакций.

**Пример действий для заказчика:**

1. Заказчик создает задание и депонирует средства.
2. Когда исполнитель выполняет задание и оно подтверждается, средства разблокируются и переводятся на счет исполнителя.
3. Если работа требует исправлений, заказчик может отправить задание на доработку. Средства будут удержаны до окончательного завершения задания.

Таким образом, обеспечивается безопасная и прозрачная система оплаты, где обе стороны защищены от мошенничества или недобросовестного выполнения работы.

4o

**Вы сказали:**

смотри мне нужэно сделать поэтапную платежную систему где первый этап это создание самой работы второй этап Поиск автора третий В работе На гарантии и завершен ка это можно реолизовать

**ChatGPT сказал:**

ChatGPT

Для реализации поэтапной платежной системы, где каждый этап привязан к определенному прогрессу задания, можно использовать систему депонирования средств (эскроу), разбитую на несколько этапов. Пример:

**Логика поэтапной оплаты:**

1. **Этап 1: Создание задания**
   * **Действие**: Заказчик создает задание.
   * **Платеж**: Предоплата или резервирование средств на эскроу-счете.
     + При создании задания заказчик резервирует сумму, но она еще не поступает к исполнителю.
     + Средства блокируются на платформе до завершения следующих этапов.
2. **Этап 2: Поиск автора**
   * **Действие**: Исполнители подают заявки на выполнение работы.
   * **Платеж**: На этом этапе оплаты нет, но средства остаются заблокированными.
     + Как только заказчик выбрал исполнителя, начинается работа, и система фиксирует начало выполнения работы.
3. **Этап 3: В работе**
   * **Действие**: Исполнитель приступает к работе.
   * **Платеж**: Частичная оплата.
     + После начала работы можно выпустить первую часть оплаты исполнителю. Это может быть определенный процент от общей суммы (например, 30%).
     + Платеж разблокируется при изменении статуса задания на "В работе".
4. **Этап 4: На гарантии**
   * **Действие**: Работа выполнена и отправлена на проверку заказчику.
   * **Платеж**: Следующая часть оплаты.
     + При завершении основной работы и её переходе на этап "На гарантии" можно разблокировать следующую часть средств (например, еще 40%).
5. **Этап 5: Завершен**
   * **Действие**: Заказчик подтверждает выполнение задания, гарантийный срок завершен.
   * **Платеж**: Выплата оставшихся средств.
     + Оставшиеся средства (30%) переводятся исполнителю после окончания гарантийного периода или подтверждения выполнения работы.

**Реализация логики в системе:**

1. **Таблица заданий (tasks)**:
   * Поля: id, title, description, price, status (этап работы), created\_at, updated\_at, customer\_id, executor\_id, payment\_stage (этап платежа).
2. **Таблица платежей (payments)**:
   * Поля: id, task\_id, customer\_id, executor\_id, amount, status (pending/paid), payment\_stage.
3. **Логика депонирования и разблокировки средств**:
   * При создании задания средства заказчика блокируются на эскроу-счете.
   * Переход между этапами изменяет статус задания и фиксирует этап платежа.
   * После каждого этапа (например, переход на "В работе" или "На гарантии") происходит частичная разблокировка средств.
   * Оставшиеся средства выплачиваются при окончательном завершении работы.

**Пример кода на PHP/Laravel:**

php

Копировать код

// Модель Task

class Task extends Model {

protected $fillable = ['title', 'description', 'price', 'status', 'payment\_stage', 'customer\_id', 'executor\_id'];

}

// Модель Payment

class Payment extends Model {

protected $fillable = ['task\_id', 'customer\_id', 'executor\_id', 'amount', 'status', 'payment\_stage'];

}

// Контроллер для изменения этапа задания

public function updateTaskStatus(Request $request, $taskId) {

$task = Task::findOrFail($taskId);

// Обновляем статус задания

$task->status = $request->input('status');

// Логика выплат в зависимости от статуса

switch($task->status) {

case 'В работе':

// Выплатить первую часть средств

$this->makePartialPayment($task, 30); // 30% оплаты

break;

case 'На гарантии':

// Выплатить вторую часть средств

$this->makePartialPayment($task, 40); // 40% оплаты

break;

case 'Завершен':

// Выплатить оставшиеся средства

$this->makePartialPayment($task, 30); // Оставшиеся 30%

break;

}

$task->save();

return response()->json(['message' => 'Статус задания обновлен, соответствующий платеж произведен.']);

}

// Логика частичной оплаты

private function makePartialPayment(Task $task, $percent) {

$paymentAmount = ($task->price \* $percent) / 100;

// Создаем запись в таблице платежей

Payment::create([

'task\_id' => $task->id,

'customer\_id' => $task->customer\_id,

'executor\_id' => $task->executor\_id,

'amount' => $paymentAmount,

'status' => 'paid',

'payment\_stage' => $task->payment\_stage + 1

]);

// Обновляем этап платежа

$task->payment\_stage += 1;

}

**Пример пользовательских действий:**

1. Заказчик создает задание и депонирует всю сумму на платформе.
2. Когда работа переходит в статус "В работе", система разблокирует 30% и отправляет исполнителю.
3. Когда работа завершена и передана на проверку (статус "На гарантии"), система отправляет еще 40%.
4. После завершения гарантии система выплачивает оставшиеся 30%.

Таким образом, каждый этап работы сопровождается частичным платежом, что мотивирует обе стороны к выполнению своих обязательств.

4o

Начало формы

Конец формы

ChatGPT может допу

ти можещ рекомендоват веб сайти работает такой етапний

**ChatGPT сказал:**

ChatGPT

Память обновлена

Да, такие этапные системы с частичными платежами реализованы на нескольких платформах фриланс-бирж. Вот несколько известных веб-сайтов, где используется подобная система:

1. **Upwork** — одна из крупнейших бирж фрилансеров, где заказчики и исполнители могут работать по системе контрактов с поэтапной оплатой. Средства депонируются и выплачиваются частично по мере выполнения работы.
2. **Freelancer.com** — на этой платформе также используется система эскроу и этапной оплаты. Вы можете разбить проект на несколько этапов, и исполнитель получает оплату по мере завершения каждого из них.
3. **Guru** — позволяет разбивать проекты на этапы с привязкой к оплате, которая выплачивается по мере выполнения каждого этапа.
4. **Toptal** — фокусируется на высококвалифицированных фрилансерах, и заказчики могут согласовывать поэтапные платежи за каждый достигнутый результат.

Эти платформы предлагают гибкую систему платежей, которая может быть полезной моделью для вашей разработки.