**ARTISTREE**

**Project Direction Overview:**

I would like to create a database for a webapp which will help hire creative freelancers and also help freelancers choose projects to work on. There are many freelancing apps out there which is a hub for all sorts of professions, but this app targets only those who work in the creative field. Such professions include musicians, sound designers, writers, lyricists, costume designers, makeup artists, jewelry makers etc. An employer can post a project online and put out RFPs on the messageboard that will contain certain skill names in the form of tags and freelancers will be suggested those based on the skills they have selected for themselves when they created their profile on the website. Freelancers can even sign contracts with their employers and keep those records on the website. Employers can also pay their recruits through the portal on the website and the website will keep track of when the payment is due and how much needs to be paid on a periodic basis based on what was agreed upon in the contract.

For example, Employer Joseph Calamaza is organizing a corporate party for an account firm and he is looking for a stand up comedian for the entertainment of the group. He posts a request for comedians who are eager to perform in front of a corporate audience at said place. He also puts down tags like comedian, corporate gala, and <name of place>. Comedian and freelancer Sandra Bullington responds to his request. Joseph looks into her profile to see her experiences and he seems impressed. He contacts her and lets her know about the details of the event and talks price with her. Once Sandra agrees, he creates a contract on the app and fills out a form where he has to mention when Sandra needs to be paid and how much amount. Those information, once signed by employing authorities, will be sent over to Sandra for her to go through and sign and send back. Both parties will have the same copy that will not be able to be edited further. Based on that, Sandra will be paid her due amount on the day it is due through the app which will already have had Sandra’s payment information. Sandra will also be rated based on her performances and Joseph can even write her a recommendation. Sandra can also rate Joseph as an employer so that more people can be helped in deciding whether to work with him or not.

This app will be called Artistree because it is like a tree with many branches of freelancing activities.

**Use Cases and Fields:**

1. **Creating an account as an employer or a freelancer - Use Cases:**

* A user enters the website for the webapp.
* They are asked to log in or sign up.
* They enter their credentials: Name, Address, Phone Number, Email, Employer/Freelancer etc.
* Account is created for them.

Significant fields for this use case are given as follows:

| **Field** | **What it Stores** | **Why it’s Needed** |
| --- | --- | --- |
| Username | This is the name using which user will login to the system every time. | Users need a credential to create their digital signature on the website. This is one of the values for that purpose. |
| Password | This is the password to secure their data. | Users need a credential to create their digital signature on the website. This is one of the values for that purpose. |
| FirstName | This is the first name of the user | This is needed to display the user’s name in any form or transaction. |
| LastName | This is the last name of the user | This is needed to display the user’s name in any form or transaction. |
| PhoneNumber | This is the user’s contact details. | For easier reachability to the user and for OTP purposes. |
| Email | This is the user’s contact details. | For easier reachability to the user and for OTP purposes. |
| Address | This is the user’s contact details. | Sending bills and storing basic contact information. |
| AccountType | Employer/Freelancer | This will denote what the dashboard will look like for the user. |

1. **Creating skills - Use Cases:**

* A freelancer will get a link on their dashboard that says Skills
* They will get a form that will let them add a skill
* They will also be able to add a type to that skill. It will be like a genre. So if a person can play guitar, their skill will be Playing guitar and it will go under the type: Music. This will help garner more music playing related jobs for them.

Significant fields for this use case are given as follows:

| **Field** | **What it Stores** | **Why it’s Needed** |
| --- | --- | --- |
| SkillName | This will be the name of the skill | If the user wants to include a skill that is not already in the system, they will be able to add it. |
| Description | This is the description if the user wants to add any. | This will be used to further describe the skill. |
| Type | This is like a genre for skills like: music, literature, dancing etc. | This will be needed to provide relatable jobs to freelancers. |

1. **Creating Job/Project - Use Case:**

* Employer/Freelancer creates a job or a project.
* They will post it with the name, venue, other supporting URLs etc for the job.
* Once someone responds to it and a contract is signed, the creator of that project will be able to assign freelancer(s) to it and the project will change its status so that it is no more publicly visible.
* Now the Project is available to both the employer and the assigned freelancer.
* Once the job is done, the employer can mark it as completed.

Significant fields for this use case are given as follows:

| **Field** | **What it Stores** | **Why it’s Needed** |
| --- | --- | --- |
| ProjectTitle | This will be the title of the project | This will be needed to display the entire project on the message board. |
| Description | This will be additional descriptions of what the project/job is like. | One the project view is opened, the description will give more details about the job/project. |
| Venue | This will be where the job/project will take place. | This will be needed if the job is to be held somewhere physical and not online. |
| Creator | This will be the name of the user who created this project. | Employer/Freelancer who created the job. |
| Tags | These will be tags that define the project | Needed to seek the attention of appropriate freelancers. |
| Status | What point of the timeline the project is now standing | This will be needed to track the progress of the job and to make sure who is having access to it. |

1. **Create Contract - Use Case:**

* Once there is an agreement between an employer and freelancer, the employer will create a contract.
* They will be able to add the project title to the contract, enter the name of the employer, freelancer, enter details of the contract and the payment details on it.
* The contract, once confirmed, will be sent to the freelancer to read and sign.
* They will send it back to the employer and a copy of both the signed contract will remain on both profiles.

Significant fields for this use case are given as follows:

| **Field** | **What it Stores** | **Why it’s Needed** |
| --- | --- | --- |
| EmployerName | Name of employer | This is to document who the employer is. |
| FreelancerName | Name of freelancer | This is to document who the freelancer is. |
| ProjectTitle | Title of project | This is to document which project the contract is for. |
| ContractBody | Description of the contract and other important details | Documentation of the details of the contract. |
| CreateDate | Date of creation of the contract | This is needed to know when the contract was created. |
| SignedDate | Date of signing of the contract | This is needed to know when both parties signed the contract. |
| PayableAmount | Amount to be paid to the freelancer | This will be needed to document and track the payment amount. |
| PaymentDate | Due date of payment | This will be needed to remind the payer when to pay the payee. |

1. **Pay freelancer - Use Case:**

* On the due date of the payment, the employer will send the money to the freelancer.
* This amount will be tracked using the payment value input in the contract.
* The employer will use the billing details that will be associated with the freelancer’s profile and send the money over to them.
* The receipt will remain in the system and users can opt for email options as well.

Significant fields for this use case are given as follows:

| **Field** | **What it Stores** | **Why it’s Needed** |
| --- | --- | --- |
| EmployerName | Name of employer | This is to document who the employer is. |
| FreelancerName | Name of freelancer | This is to document who the freelancer is. |
| PayableAmount | Amount to be paid | This is used to keep track of the payment. |
| PaidAmount | Amount paid | This is used to keep track of how much is paid in this iteration and how much is remaining |
| DueAmount | Amount remaining | This is used to keep track of how much money needs to be paid after this iteration |
| ContractId | Link to the contract | This is used to link the contract with the payment details. |
| BillingDetails | Billing details of freelancer | This is needed to send over the payment to the freelancer. |

**Summary and Reflection:**

Artistree is a web application that will be used to provide jobs to creative freelancers and for employers with jobs to post them to get the best persons to the job for them. The database assisting the webapp will be used to store basic details of employers and freelancers along with their billing details and even how much rates they offer for a job. Freelancers will be able to add skills in order to be notified about jobs posted related to their skills. When an employer posts a job, they will be able to add tags to the post, this will help garner more attention from freelancers suitable for the job. Once agreed upon, a contract may be created between the employer and freelancer and based on that, the employer may pay the freelancer too. The system will keep track of how much is paid and how much is due based on the total payable amount.

As I was designing the use cases, I did feel like this is going to get much bigger than I imagined but I hope through the future iterations, I will be able to get a hold of the design.