|  |
| --- |
| **Project Report** |

**Atharva Patil Nitin Vinod**

Department of Computer Science Department of Computer Science

University at Buffalo University at Buffalo

[*atharvap@buffalo.edu*](mailto:atharvap@buffalo.edu)[*nitinvin@buffalo.edu*](mailto:nitinvin@buffalo.edu)

**Abstract**

A Dapp where users can delegate small but important daily tasks to professionals in exchange for a denomination of ERC tokens. Users(customers) will be uploading small tasks and other users can pick up those tasks in exchange for some ERC tokens. The price that the customers incur for the services will be paid by using our ERC token. The main goal is for us to satisfy all these needs by connecting these customers to professionals on our platform. The price that the customers incur for the services could be paid by using our ERC token. We wish to fix the issues pertaining to gender/racial bias in online platforms. In a decentralized system, participants will be prevented from forming stereotypes about the professionals available on the online platform. This measure will make sure that members belonging to a specific community will not be given any unfair advantage.

1. **Use Case:**

**Diagram

Description automatically generated**

**Description of use case diagram:**

* There will be 2 users: Assignor and Assignee. The Assigner will be adding a new task on the website, which will be picked up by the Assignee.
* Once the task is picked the token amount can be sent from the Assignor to the Assignee.
* Both the users will be able to see the number of jobs completed by the Assignee and the rating of the Assignee.
* The users will not be able to see the name/gender or any form of identification of the other user, hence avoiding any kind of bias while assigning a task

1. **What is the idea of the project?**

* The basic concept of the project is to create a task publishing application which will be free of any bias and users will be able to get their tasks done.
* Tasks like making a small presentation can be published on the website, the Assignees can pick up the task and complete it for a certain number of tokens.
* Using IFPS the assignor can upload specific instructions or files for the assignee. A hash or a link of the file can be shared to the assignee where details of the task can be found. However, the IPFS application has to be installed and setup on both the assignee and the assignors computers.

1. **Why use our tokens over normal transaction using USD?**

* The main advantage of using a cryptocurrency or a blockchain token is that it does not have to go through a third party.
* Using a third-party financial institution like PayPal has its own disadvantages like data insecurity and prone to hacking.
* Using a blockchain token also ensures a smart contract can be run for a transaction which is not possible when using cash or digital currency (Eg: USD).

1. **How is our website different than another task publishing website?**

* The other task publishing websites do not use blockchain technology, blockchain has its own advantages which makes the websites significantly better than any other website.
* Few websites may use blockchain, we differentiate ourselves by focusing on using the decentralized ID to keep the names and identify of the user hidden which makes our website free from any kind of racial/gender bias created by users

1. **Why use blockchain technology in our project?**

* In large organizations where a lot of data is generated, the data gets stored at a single source, which poses high risk
* Using a third-party financial institution like PayPal has its own disadvantages like data insecurity and prone to hacking.
* Using a blockchain token also ensures a smart contract can be run for a transaction which is not possible when using cash or digital currency (Eg: USD).

1. **Contract Diagram**

**Text

Description automatically generated**

**Description of the Contract Diagram:**

**Contract 1 TaskToken.sol:**

* Function TotalSupply(): Used to get the total supply of the token
* Function balance(): Used to get the balance of tokens in the selected account
* Function transfer(): Transfer tokens from one account to another or vice versa
* Function approve(): Gives permission to transfer funds within a specified limit from a selected account to another account.
* Function allowance (): Used to provide the limit on the funds to use in approve() function.

**Contract 2 SmartContract.sol:**

* Function job\_completed(): Transfers the money(token) to the assignee from the contract after job is completed.
* Function job\_not\_completed(): Transfers the money(token) back to the assignor as the job is not completed.
* Function add\_job\_details(): Adds the job description, job ID and any instructions using IPFS.
* Function get\_job\_details(): Returns the details of the job.
* Function add\_assignee\_details(): Takes input for the ID of the assignee and the Job\_ID.
* Function get\_rating(): Returns the rating of the assignee after the job is completed
* Function no\_jobs\_completed(): Returns the total number of jobs completed by the assignee.

1. **Sequence Model of project:**

**Flowchart and sequence diagram**

Diagram, box and whisker chart

Description automatically generated

**Diagram

Description automatically generated**

1. **How to run the code:**

**Important terms to understand before start:**

* **Assignor: Person who puts the task on the platform**
* **Assignee: Person who completes the task from the platform.**
* **Job\_Id: Unique ID for a job.**

1. First, we need to connect the smart contract to React.
2. Deploy the contract on Remix.

Graphical user interface, application

Description automatically generated

1. Copy the ABI (application binary interface) from the Solidity compiler tab.
2. Copy the address from the deployed contract section.

A screenshot of a phone

Description automatically generated with medium confidence

1. Paste the address in taskplatform.js as the value for const address
2. Paste the ABI as value for const ABI.
3. Use the contract address and get 1500 tokens (total supply) into the contract owners account and send some tokens into 2 other others for assignor and assignee to implement the project.
4. Start app using npm start on the terminal, and login into the metamask wallet.

Graphical user interface, website

Description automatically generated

1. Go to Account of your Assignor and add the tasks and give IPFS link, task name and cost.
2. For our example we will be adding 3 tasks

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

1. We can see that the balance in account 2 has reduced from 150 TT to 90 TT as we added 3 tasks for 10, 20, 30 TT each.

Graphical user interface, text, website

Description automatically generated

1. We can view the tasks on the website.Graphical user interface, website

   Description automatically generated
2. Go to Account 3 so we can start beginning taking tasks as a assignee.

Graphical user interface, text, website

Description automatically generated

1. Now start picking up tasks using assignee account.

Graphical user interface

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

1. Now go to Account 2(Assignor account) and use job completed and give a rating.

job\_id = 10 and rating= 5

job\_id = 11 and rating= 3

USE JOB\_NOT\_COMPLETED for job\_id = 12

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

1. Now we will see that Account 2 has a balance of 120 and Account 3 has a balance of 180 as the task 10, 11 got completed which was confirmed by Account 2 and hence money was transferred from the contract to Account 3 while for task 12 money was transferred back to Account 2.

Graphical user interface, text, website

Description automatically generated

Graphical user interface, website

Description automatically generated

1. Now we can view the average rating of the assignee using their address.Graphical user interface

   Description automatically generated
2. Now we can view the average rating of the assignee using their address.

Graphical user interface, website

Description automatically generated

1. The End