WEB-3

POLL

A Polling Smart Contract System using Blockchain Technology

Submitted by: Aadarsh Kumar Rauniyar

6/12/2022

Submitted By:
Aadarsh Kumar Rauniyar
C7227242

Submitted To:
Saroj Sharma
Supervisor

I. Abstract

Poll is a block-chain-based polling/voting tool that combines web2 functionality with web3 authenticity. It uses the web2 website platform, but links to a smart contract that works as the backbone for all transaction votes, keeping all data safe and public and unconnected to any particular company. Block-chain is a decentralized network for sending and receiving data in the form of transactions. These transactions are kept in blocks of the block-chain, which may then be accessed using block-hash. With a centralized database, an online voting software was never perfect on its own. Block-chain technology solves this significant flaw in existing online voting systems by logging all votes as transactions and decentralizing the system, which takes authority away from a single party in power.

II. Student Declaration

I, the undersigned Aadarsh Kumar Rauniyar student at The British College (B.Sc. (Hons.) Computing) hereby declared that the project work of “A Polling Smart Contract System using Block-chain Technology” presented in this document is my word and has been carried out under the supervision of Mr. Saroj Sharma of The British College. The materials contained in this report have not been submitted to any University or Institution for the award of any degree.

Aadarsh

Signature

Date: 10/06/2022 Student: Aadarsh Kumar Rauniyar

Place: Thahity, Kathmandu C7227242

III. Table of Contents

Contents

IV. List of Figures/Tables

V. List if Appendices

VI. Acknowledgements

I would like to express my sincere gratitude towards my supervisor, Mr. Saroj Sharma, who has blessed me with his guidance and support towards completion of this project. I’m very grateful to the British College and my Supervisor for allowing me this opportunity to explore the block-chain technology as a module project. I am filled with gratitude to recognize the support of my Supervision for allowing me to explore a new field of technology when I had no expertise prior to project start.

1. Introduction

Voting has been a part of our society since very long, it has been widely practiced and accepted way of decision making for communities and democratic believers. With time, the way of polling has come to change and has adopted the new technology. Before, polls and voting were done in papers and conducted manually, this process was fit for the days before internet became a thing. Online voting's were finally possible but wasn't as authentic as votes done manually on papers. Often times, the data on the internet hold of potential of getting hacked and rigger, causing the voting process to be not authenticate and completely destroying the morals of voting. Block-chain is the new technology introduced to us as the web3, an upgrade to web2. Block-chain is a completely decentralized platform that sends data to and from a system in the for of transactions. These transactions are saved in the blocks of the block-chain which is then accessible with the help of block-hash. All the transactions are recorded and saved on the block-chain and can be retraced back when needed. The internet is slowly upgrading to web3 and all existing applications will slowly be converted into d-apps. An online voting app was never perfect on its own with a centralized database. The authenticity of the voting app is as good as the company conducting it, and there is need of a significant amount of trust for the voting results to be fair. Block-chain technology removes this major issue that the current online voting apps have by recording all votes made in form of transactions and decentralizing the system taking control from a single governing party.

- **Research**

  The risk on electronic voting system is very critical and possesses a significant risk on the overall system no matter how small the exploit or problem is. This issues is getting generalized in the growing internet world but the problem still exists and is truly holding back e-voting systems to overtake the manual voting system. (Weldemariam, 2010) Block-chain has helped with its features in many sectors like finance, medical, real estate. The desired voting system that has security, accuracy, scalability, transparency is achievable through the block-chain technology. (Shaku , et al., 2019) Block-chain is the solution to making e-voting systems because of the transparency is offers. The use of smart contract is to be done to make the desirable Voting system.

- **Product Background**

Poll is a polling/voting app based on block-chain that bridges web2 functionality with web3 authenticity. It makes use of web2 website platform but connects with a smart contract that acts like the backbone for all the votes made in transactions keeping all records secure and public unassociated to a single organization. This smart contract made using the solidity program focuses on making new poll commands as well as voting on them. The scope of this app is very vast and can be utilized by many parties and organized. The app is focused for use of a small company where they can use is to make some democratic decisions for the company based on inputs from the colleagues and company team workers. This will allow the company to take active decisions with the upgrades and future plans for the company and actively work on the betterment of the company. The trust block-chain and smart contract provides is the highest when it comes to online voting and this increase more accuracy and interest on the decision to be made. The process of voting is also simplified helping in the increase of active voting members helping the company to reach better stats and collect more data. Smart contracts can be deployed on many available block-chains, all have their own benefits and drawbacks. Ethereum is the most popular and widely used block-chain for smart contracts and D-apps but comes with a drawback for having higher gas fees for any transactions to be made. This causes some issue when it comes to polling system as the target is always making it free for the members to vote, helping in increase of total votes. While Ethereum has high gas fees there are other major block-chains such as Solana that provides same services as Ethereum but has close to no gas fees. Services like these are what makes the product a cost friendly experience and more practical in a working environment.

2. Review of Literature

Polling system is a voting system made using block-chain technology demonstrating the transparency functionality of the block-chain and smart contract. There are decisions to be made in a company. The discussions could be for various topics and for few a voting among the responsible members can be initiated. This voting process can be made easier with the help of block-chain based polling system. This product focus on the sentiment of the polls being made and allows the user to make their votes and displays the live results on the webpage reflecting the sentiment of the total votes in favor or against.

- **Project scope and Requirement**

  Services:

  a. Easy UI experience for the end user
  b. Product development thorough web3 elements and software
  c. Testing of product for checking overall quality
  d. Running the product and maintenance

- **Desired product capabilities:**

  a. Working smart contract with needed functionality
  b. Wallet connection on the webpage
  c. Adding votes on poll
  d. Voting eligibility checker
  e. Presenting poll results

The comparison for the project are mentioned below along with their data sources.

a. **E-ballot**: This is the leading e-voting platform that is widely famous for its security and reassurance. This platform is used for many voting purposes and is widely capable for multi-purpose voting system. This app features a login functionality and makes profile of the user. The app here is stored centralized. E-ballot is one of the best e-voting system out there but the major difference between it and the product is its lack of decentralized in nature. The product is based on block-chain and has the transparency of this technology.

b. **Electronic Buddy**: This is a more user friendly app for electronic voting and features free use for up to 20 users. To be able to use it in a more working environment, it comes with a cost. Like the E-ballot, Electronic Buddy is also a centralized system and doesn’t make use of the block-chain like the product, polling system.

c. **Polys**: Polys is a block-chain based online voting system. Polys makes use of the decentralized system and allow a login portal for the users where users can create a profile and engage in voting. The voting process is held and results are kept secretive until the end of voting. The polling system is also based on block-chain technology but here the voting results are shows immediately on each votes being made, which acts as a proper sentimental review on the decisions to be made by the company.

d. **Voatz**: This is a block-chain based mobile voting system. It features a mobile app for all the voting requirements. A mobile app is very useful as it’s more accessible and convenient for users around the globe. The polling system, is based on a web-page app and cannot be as accessible as the mobile app.

Overall, the unique aspect of the polling system is its result showing mechanism. The live update of the result shown after each vote and shown in percent form to indicate an overall positive or negative response on the poll being made is one of the crucial features of the product. The smart contract involved in the product is also crafted uniquely to meet the purpose of the product to be able to make votes on the available polls as well as allow the admin/ the creator of the smart contract to add new polls on the system. This small polling system can be best used as a company sentiment analysis where a company can create polls on different matters to take opinions and suggestions from its members.

3. Review of technology

Technology has grown vastly in the past decades, and humanity has never seen such heights of technological development. The major technology that is used in the product is Block-Chain. The product is completely based on the block-chain. Firstly, talking about block-chain, it was first introduced in October of 2008 (Iansiti & Karim, 2017) as a proposal of Bitcoin, a virtual digital coin existing between peer to peer networks that replaces the current monetary economy overthrowing the centralization of banks. Block-chain is like a collection of distributed database where transactions take place between two parties with secured and recorded records without the involvement of third parties. Block-chain has introduced the world a new decentralized way of handling data and has made as such removing any requirement and needs of a central governing body to run a system. A system truly governed over multiple decentralized units that overcomes the common centralized system shortcomings such as single point variability and trust issue, has been made possible with the help of Block-Chain technology.

Block-chain technology is being vastly adopted in the new decade and a lot of larger companies have started to make use of this technology. The use of the technology has found its way in to food industries, health care, Property records, Smart contracts and supply chains as well as voting. In this project, the scopes and functionality of smart contract and voting system are tested on the Block-chain.

A Smart Contract is a set of programmable codes that codes the structure of a system for transaction confirmation. It provides a set of rules and hard coded statements that when fulfilled will only allow any transaction to be made between parties. In simpler words, it’s a digital contract made on the block-chain between the sender and receiver that has a set of requirements which when fulfilled will allow the transaction between parties even in absence of one or the other. “Smart contracts gave network automation and the ability to convert paper contracts into digital contracts.” (Shafaq , et al., 2021) A smart contract can have as many number of conditions as required by the parties which are all written in form of simple if/then code form. Smart contract have a lot of benefit in the block-chain eco system with automating the system being the most important out of all. Smart contracts are very secure and the conditions on contract cannot be manipulated by any parties thus proving to be a realizable security major on various systems. The accuracy, efficiency and speed of transaction made with contract are not compromised and as soon as the conditions are met, the transactions are recorded on the block-chain. (Anon., n.d.)

In the product, the use of smart contract has played a major role in making the outcome of the product. The whole polling system is built over the contract that acts as the backbone of the system ensuring that only valid transactions are being made on the system.

….

4. Methodology

The set of rules and guidelines defined for a development of a research project is called Methodology. Methodology are required for the identification of patterns and for better prediction in the process of research. (Little, 2014)

The ways of the development approach to a research is very important for the better understanding and smooth process. The steps to overcoming a problem it to be identified early hand so there can be a systematic approach for the completion of the research. There are multiple types of approach when it comes to Methodology, some are better than the other in different aspects while none being inferior to the other. All types of methodology have its own perks and shortcomings. The most widely adopted methodology that has been introduced since the early days of app development is the Waterfall Methodology. In this, all the processes of completing the research is identified early hand and the development is made completing each divided process one after the other in a fixed pattern. The best part about this methodology is that the product development can most of the time be guaranteed because each step is to be cleared with perfection. The drawbacks of this methodology are quite significant to the modern days of research because of the fact that there is no going back in the waterfall methodology. And sometimes the process can get stuck on some part that is being blocked by some unavoidable circumstances. In this condition, the development cannot proceed and will take as long time as the circumstances cannot be fulfilled. In modern days, this methodology is slowly being used less and less.

The other famous methodology type is prototype methodology which supports the process of trial and error and plans ideas and processes with a prototype in mind. In this framework, the process of approach is divided into phases that focuses building a prototype before developing the actual application. This approach to methodology is very critical in engineering field and has been proven to be very cost friendly among researchers. The prototype build is focused on trying new deigns because it is capable of reducing work and cost on the actual project is things go south. The only drawback of this methodology is that is requires the full understanding of the research and product while making prototypes before the completion of the product.

When time constrain becomes of factor of development progress, there is a methodology appropriate for this type of approach to meet the deadlines. Spiral Methodology is a risk-driven method in approaching development. (Software, 2020) This framework makes use of multiple methodology and its aspects in a single project in the efforts to meet the limited deadline. The results this methodology produces has a high risk in quality as it is based on the principles of high-risk high-reward. As said by Mark Zuckerberg himself, “The biggest risk is not taking any risk... In a world that is changing really quickly, the only strategy that is guaranteed to fail is not taking risks.” This methodology is widely used and especially popular among nonprofessional workspace.

Another popular methodology is Agile Methodology. This is one of the popular choices among other methodology mentioned earlier. This method focus on the distribution of project in multiple different parts and each part can be accomplished separately which later is assembled for the completion of the project. Agile Methodology is widely in practice in the modern development environment especially when many hands are required in the making of project. The objective of this methodology is to identify the major aspects of the project and divide it into phases where each phase is handled by its relative expert team.

The Polling system was made with the use of Waterfall methodology and the whole process was divided into these phases:

i. Researching the Required Components

   Firstly, research was made for all the components related with block-chain development and web3 aspects. The possibilities of block-chain use was researched and components required for the development of the polling system were analyzed. Need of smart contract and tool for deployment of smart contract was discovered and multiple ways were researched to find the required tools for the product. Front end frame works were also explored during the research as well as some brief knowledge of connecting a smart contract with a front end was researched.

ii. Making of Smart Contract

The main backbone of the product lies on the Smart Contract. Firstly the possibilities on the smart contract was explored and the required minimum procedure was identified for the required polling system that allows the addition of polls and vote on polls. The smart Contract was made and tested using the Remix local block-chain environment as well as the local block-chain provided by the Hardhat. This phases was fairly easier for completion as it holds basic knowledge programming codes.

iii. Deployment of Smart Contract

Smart Contact is as good as just simple if/else codes if not deployed on the block-chain. For the deployment of smart contract, the various possibilities were explored and researched. Local Block-chain were used for initial testing of smart contract and later the use of official Test networks powered by bigger block-chains were made use. Deploying Smart Contract on the block-chain comes with a fee which is to be paid in crypto, so to overcome this issue and make this process money less, the use of test network are used. The test network are widely famous among web3 developers and it is wise to test any contract on a test network before deploying it to the actual main network. The Test network used in the product is Mumbai which is powered by the Polygon. After the smart contract was deployed on the Mumbai test net, the contract was officially ready for interactions and valid transactions.

iv. Making of Webpage

Smart Contract is great but for the interaction of user with the smart contract in a friendlier UI (User Interface) there is need of a font page. A webpage acts as a medium between the user and the software providing a platform for interaction. There are many framework available for making the front end. Some famous frame work discovered were React and Next JS. React was used as the base of the front end of the product. All the design and CSS were completed through the libraries provided by the React.

v. Connecting Webpage with Smart Contract

Lastly, the methods of connecting the webpage with smart contract, both of which was created in the earlier phases, was identified. The use of Moralis and its tools were made use of for this task. The server to extract data straight form the block-chain was also provided by Moralis. The smart contract functionality of voting on the available polls was connected with the front end and now the user can trigger the transaction on the block-chain straight though our one page app.

Here, every phase was completed before the next phase and additional research was made in every phase for the complete understanding of the final product as well as the possibilities these components provide for the future of the product. The whole process was completed one by one following the Waterfall Methodology. As expected, the process did have to face some major drawbacks that the waterfall method has. When product was being designed in the front end phases, due to compatibility issues later explored that the smart contract deployed has to be deployed using other program. Likewise, during the time of connecting smart contract with frontend, there was need of making changes in the front end side which was totally unsatisfying and time consuming.

Overall, the use of Waterfall Methodology has been effective for the whole development journey. Other methods of methodology, especially Agile Methodology would also work effectively if the product were to be assigned to a team.

5. Product Design

The product was designed with the help of many resources that includes web2 as well as web3 components, mainly web3. The backbone of the product which is Smart Contract, was made using Solidity. Solidity is an object-oriented high level programming language specially made for designing and writing smart contract. There are many platform to work with solidity, where Remix is the most popular choice as it provides its own block-chain for testing the contract while being in the phase of construction. This crucial feature allows the coder to test the functionality of the smart contract and ensure the final contact being made is all audited before implementing in the main block-chain.

Presenting below the flow of the product design as well as the user interaction for better understanding moving further.

Data flow Diagram

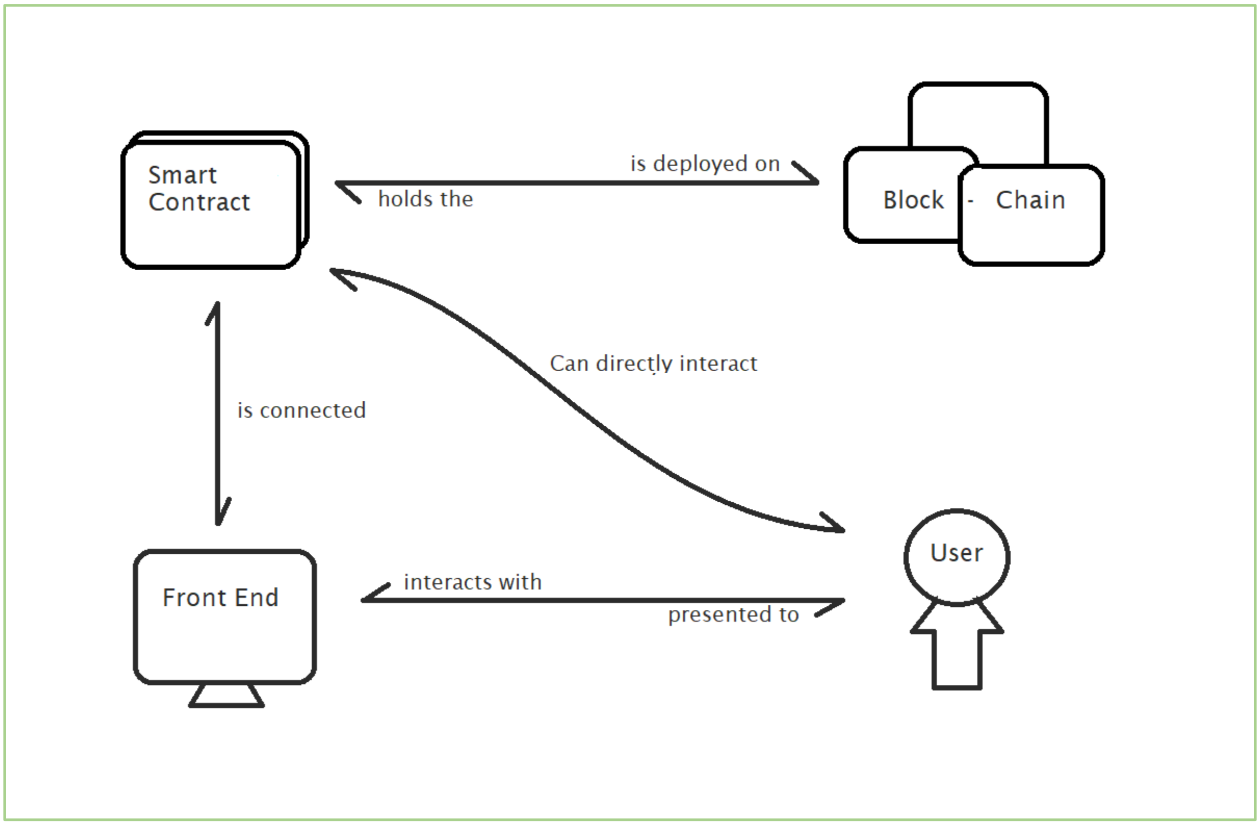


Figure 1 : Data Flow Diagram

This diagram explains the main backbone flow of the product and simply shows the overall connection between elements in the system. Here, it can be seen that the smart contract is deployed on the block-chain, and users can interact with the smart contract to add inputs as they desire. It is also shown that the front end exists between the user and contract to make the user experience smoother by providing an interface for interaction with the smart contract.

Use Case

Use case is a part of system analysis that is responsible for identification and arrangement of system requirements that lays out all the possible interactions (between user and system or admin and system) possible in the system presented in figure form. (Brush, n.d.) The use case for the product was simple as there is not much for interaction. The different use cases are as following:

User use case

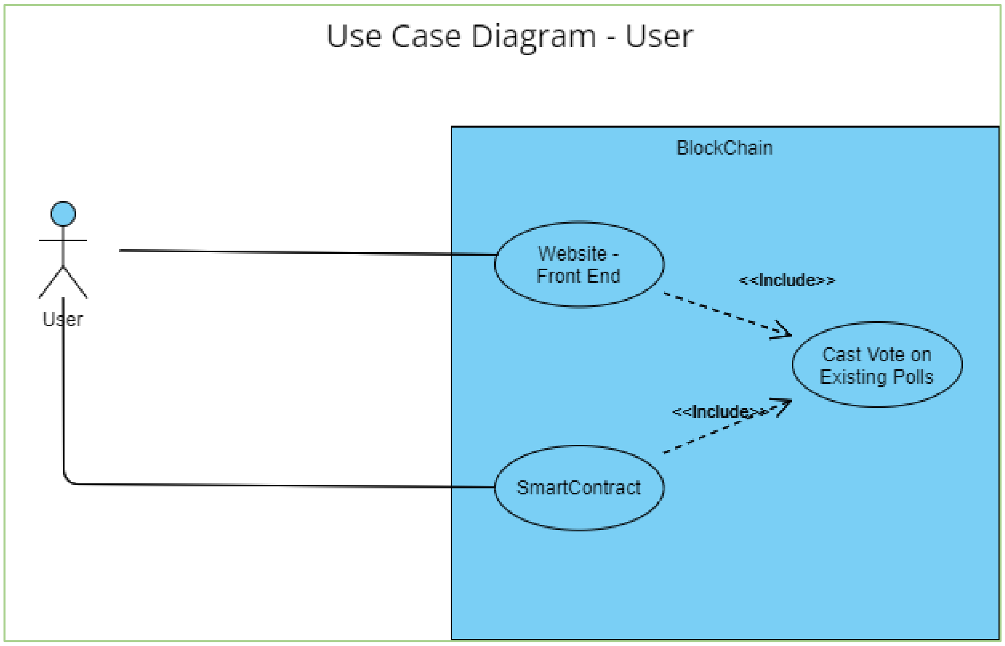


Figure 2 : User Use case Diagram

In the above diagram, it can be seen that the user can interact with the smart contract or the front end of the website for casting vote on existing polls on the system.

Admin use case

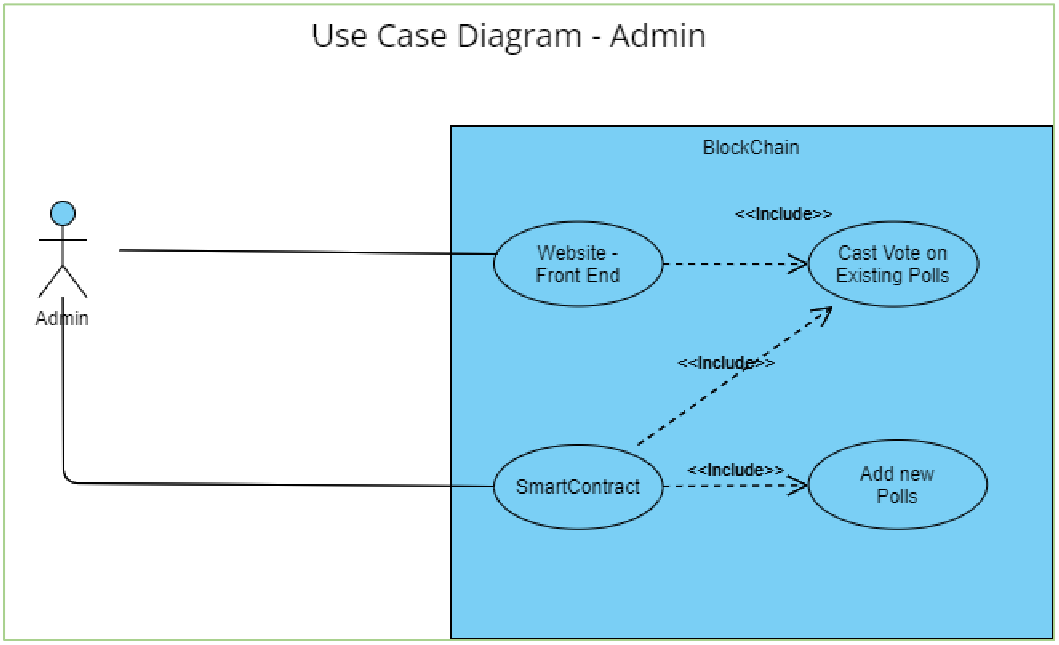


Figure 3 : Admin Use Case Diagram

From the above figure, it can be clearly observed that the admin is able to cast vote on existing polls through contract as well as front end. And the admin can add new polls through the help of contract.

System Use Case Diagram

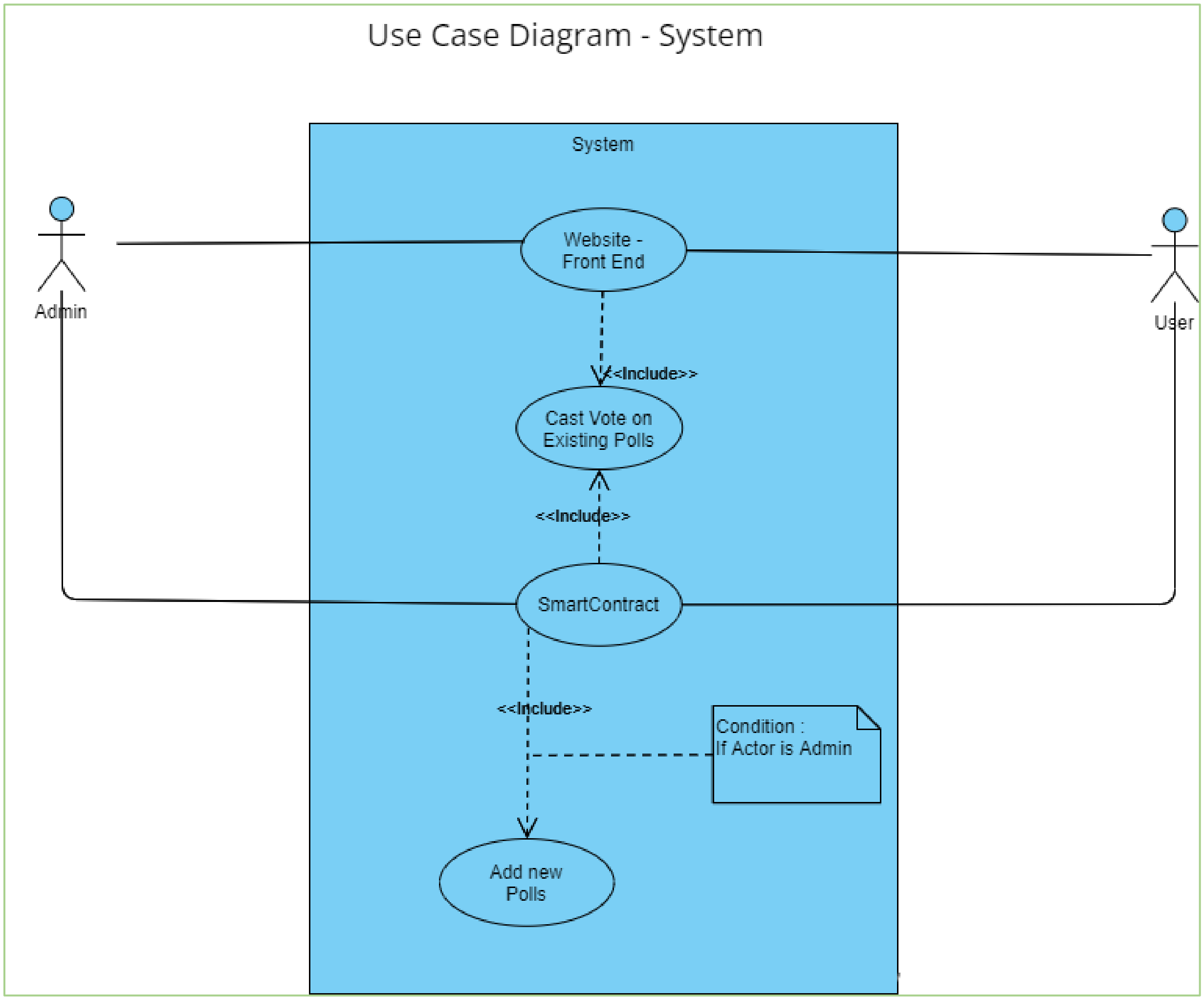


Figure 4 : System Use Case Diagram

This is the system use case diagram where it shows the overall interactions made from user and admin as collective in the system. Here, it shows that only the admin is able to add on the polls thus keeping the system under control and easier to maintain.

Design Process

After the constructing of the audited smart contract, the contract was then opened in IDE, Visual Studio Code. Visual Code Editor is a code editor software that is mainly used for crafting and debugging new web as well as cloud apps. Solidity compatibility was added on the IDE and for deploying the contract, Hardhat was used. Hardhat is an Ethereum development environment that offers the coder a local block-chain for deploying the contract as well as supports deployment of contract on other all kinds of block-chain network such as ethereum mainnet, polygon, rinkyby. Firstly, the contract was deployed on the hardhat local block chain for re-audit of deployment compatibility. After that, the polling contract was deployed on the Mumbai network, which is a test network powered by the pologonscan block-chain. Test net block-chain are the networks that is separate from the actual main block-chain which is helpful for testing of contract and ensuring the security. Crypto-currencies on Test network are faucet currencies and cannot be swapped to the main block-chain. After deploying the contract to the Mumbai network, the product can already be accessed for interaction but the UI available isn’t the most sound.

Moving on to the front end part of the product. The front end was created with the help of React. React is a JavaScript library that mainly focuses on making front end. (Platforms, n.d.) The unique feature of React is the ability of making single-page user interfaces where only needed portions of the webpage is changed as the user interacts keeping the whole page as one without the need for requesting the server a new page load. This makes the user interaction very smooth and more convenient and all in all a better user experience. The overall design on the front end was made using React and CSS components. CSS (Cascading Style Sheets) is a style sheet language that focuses on the styling part of the webpage, a must-component for the front end. After the completion of webpage design, lastly the connection of Webpage and Smart Contract is to be made. This was achieved with the help of Moralis, it is a workflow that provides a user powerful components for designing web3 D-apps (Decentralized Applications). The product has made use of majority of the services that Moralis has to offer for the connection of web page with Smart Contract. The web3 server, wallet connect features and contract database that directly pulls data from the contract transaction history are the few features used that Moralis has to offer.

The product is designed is a way that the user experience in understanding the app is very easy and minimal effort is required. The user is presented with a single page app where the existing polls are presented which features two button for voting up or down as the user choice. Once the decision is made and the button is clicked, it then triggers the smart contract which then validates the validity of the user address that is currently connected to the page. If the user has already voted on the poll once, then the smart contract won’t allow any further transactions to pass. This was achieved with the smart contract code that was written for this polling system. Also, the specialty of the product design is that is shows live data fetched directly from the smart contract and displays the sentiment of poll in percent. If the total number of up votes on the poll is more than half then the result of the poll on the app will be presented in a green theme clearly mentioning the total percent of the overall sentiment. Likewise, when the number of down votes on a poll is higher than the poll on the app will be presented in a red theme. This change is directly connected with smart-contract live feed, and will updated live as new transactions are made through the smart contract.

Like many other Apps, this app is also not perfect and comes with few drawbacks and also possible future improvements. For now, only the admin of the app, which is the contract maker can only add new polls in the webpage. This is done to ensure the security and authenticity of the app reducing any manipulative actions. The only drawback that exists is that the admin has to use the smart contract to add new polls, and when new polls are added the front end side has to be readjusted as this process has not been made automated yet. For now, the user and the admin can vote on newly created polls though the contract only, causing some lack of user friendly environment in the system. Other than that, the live graphical update of the poll result is very handy and easy to understand for the user, making the app more user friendly.

6. Software Requirement Analysis

In software development, understanding the needs of a customer and analyzing the product accordingly has become a challenging difficulty. In the context of bad software requirement analysis, it can cost the project its time, money and quality. (Senay & Resul , 2018) In the initial phase of the product as well as the final phase, the requirement analysis should always be kept in check to achieve the desirable product.

Voting system apps have already been in the current marketplace and the internet. There are multiple places where the voting system are used. Could be for an election of the head manager, or could be for to decide whether the community things ideally about the future plans made by the head organization. Whatever be the reason, a system of voting on two or more polls have always been required in the community. Normal web2 polling system exists and are widely used in the current society. But with limitations and central in nature of the web2 technologies, these polling system faces a major challenge, Validity. Validity of votes that are made is as good as the company running the voting system. There are major security concerns on the authenticity of the votes. As we know all web2 software make use of a database that stores all the information's of their customers and consumers in a locally central place. This is a very major issue when it comes to the safety aspect of the votes. A single point storage system if compromised will cause leak of all the information stored by the company.

Voting and polls are sensitive information and can easily be targeted by malicious parties for their own benefits. The only thing guaranteeing the authenticity of the votes being made is the company that holds the voting session itself. As all the vote data are stored by that company privately, the concerns of trust arise and suspicions on people with high power in the company arises as well. Voting is a sensitive matter, it shouldn't be handled by a single organization or company. Decentralizing the system while upgrading the security is one of the solution for this.

Here comes the need of block-chain, decentralized system, the major aspect of the block-chain. Block-chain is the backbone for all decentralized applications and software. Block-chain technology fixes the required need of a voting app, making it completely decentralized and out of control from a single parties. Doing this will provide authenticity to the voting system and will also make the system much more trust worthy as all the transactions on the block-chain are recorded and accessible by anyone. The security that the smart contract provides to the voting app is very assuring as the code for the smart contract are also visible to everyone and thus any eternal auditing is welcomed at any times increasing even more reliability and trust to the system.

Though smart contracts are very reliable and trustful, they don’t come with the best user interface for interaction for the common user. Thus, there is need of a user friendly environment to interact with the smart contract. Also the UI to check and get the result on the block-chain isn’t the best thus requiring a better representation of the results of the poll. The product is designed in such a way that the system requires minimum interaction from the user and shows the result of the votes in a much graphical manner. Created polls will be displayed in the front page of the app representing the current score of the poll and two simple buttons for voting up or down interaction. On interacting with the buttons will trigger transaction which will cause change in the poll result visually.

The software Requirement according to the initial project plan are as follows:

Function Requirement

Function Requirement represents all the product’s features and design functionality that is needed for the successful interaction of the user with the product. Here, according to the initial plan, the product should focus on developing a smart contract for the backbone of the system using solidity. Also the product should successfully demonstrate its usability in a working environment for a small company. The polling system should have an eligibility checker as well as transparency of polling and votes. And lastly have a small website to deploy the smart contract.

REQUIREMENT ID	DESCRIPTION	MOSCOW RULE
A1	**Functional Requirement**	
A1-01	A smart contract for polling using solidity	**M**
A1-02	Demonstration of polling system for a small business organization.	**M**
A1-03	Polling Eligibility checker	**S**
A1-04	Transparency of polling and votes	**S**
A1-05	Clarification of how smart contract is secure	**S**
A1-06	Documentation of the Solidity code	**C**
A1-07	A simple front end website to deploy the smart contract	**M**

Non-functional Requirement

The system requirements that focuses on the usability, manageability, scalability and such needed attributes for the product are referred to as the non-functional requirements. Here according to the initial project plan, the must requirements were to have a secured polling method using smart contract and as well as to have a front end website for the interaction with smart contract. Providing information about the block-chain and smart contracts are also one of the must functionality. The method of voting is to be made transparent as well as the eligibility of voting are to be clarified. The goal is to make it a user friendlier product as much as possible.

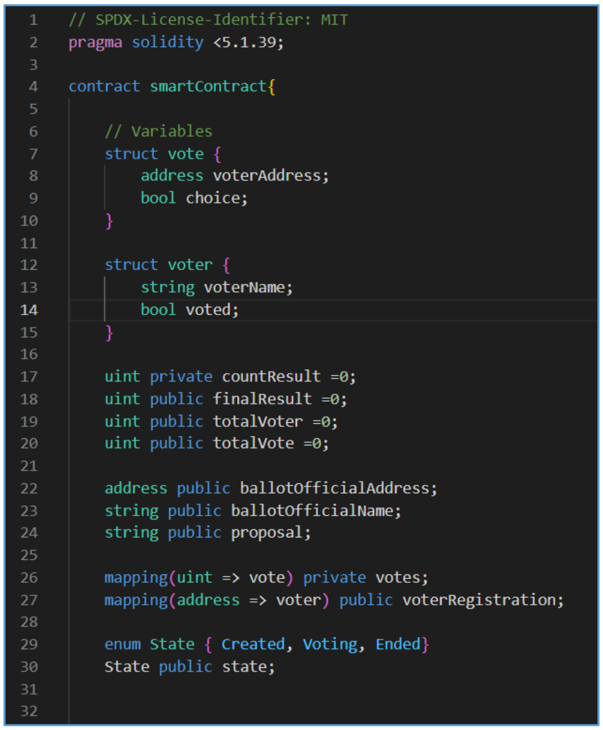
REQUIREMENT ID	DESCRIPTION	MOSCOW RULE
B1	**Non-Functional Requirement**	
B1-01	Have a secured polling method using Smart Contract	**M**
B1-02	Provide detailed information about block-chain and smart contracts	**M**
B1-03	bring transparency in polling	**S**
B1-04	Show eligibility for voters on polling	**S**
B1-05	user-friendly product	**C**
B1-05	A front-end website for interact with smart contract.	**M**

7. Implementation and Testing

In any development process implementation and testing of the product is absolutely crucial before finalizing it. Likewise, in the product the possible implementation process was researched and lastly a simple brief process was discovered making it as user friendly as possible. Likewise proper needed testing was also performed for ensuing smooth user consume.

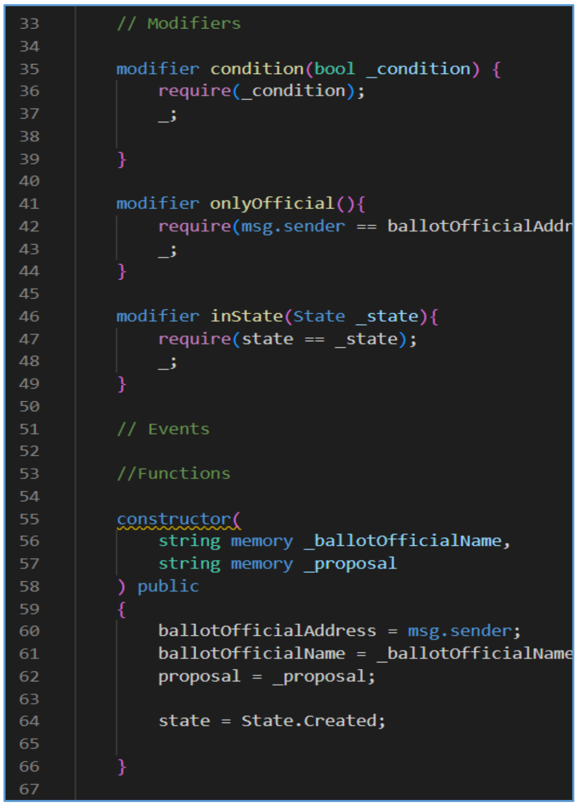
i. Implementation

The implementation process started from making of the smart contract.



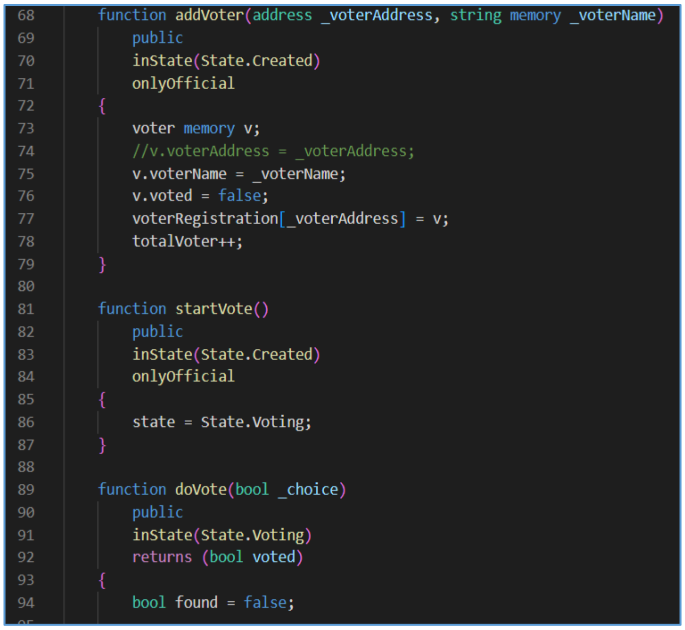
```
// SPDX-License-Identifier: MIT
pragma solidity <5.1.39;

contract smartContract{

    // Variables
    struct vote {
        address voterAddress;
        bool choice;
    }

    struct voter {
        string voterName;
        bool voted;
    }

    uint private countResult =0;
    uint public finalResult =0;
    uint public totalVoter =0;
    uint public totalVote =0;

    address public ballotOfficialAddress;
    string public ballotOfficialName;
    string public proposal;

    mapping(uint => vote) private votes;
    mapping(address => voter) public voterRegistration;

    enum State { Created, Voting, Ended}
    State public state;

```

Figure 5 : Smart Contract 1



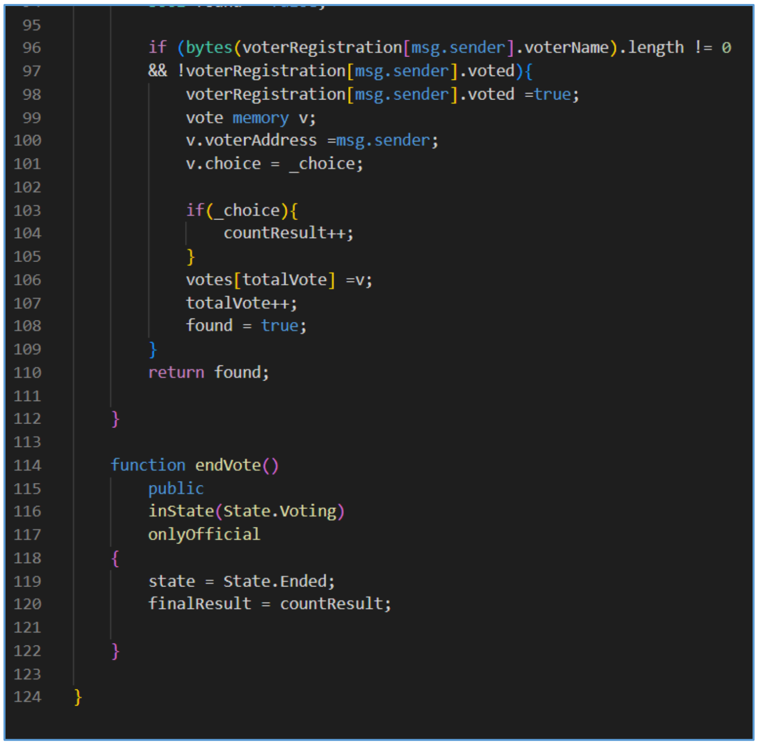
```
    // Modifiers

    modifier condition(bool _condition) {
        require(_condition);
        _;

    }

    modifier onlyOfficial(){
        require(msg.sender == ballotOfficialAddr
        _;
    }

    modifier inState(State _state){
        require(state == _state);
        _;
    }

    // Events

    //Functions

    constructor(
        string memory _ballotOfficialName,
        string memory _proposal
    ) public
    {
        ballotOfficialAddress = msg.sender;
        ballotOfficialName = _ballotOfficialName
        proposal = _proposal;

        state = State.Created;

    }

```

Figure 6 : Smart Contract 2



```
    function addVoter(address _voterAddress, string memory _voterName)
        public
        inState(State.Created)
        onlyOfficial
    {
        voter memory v;
        //v.voterAddress = _voterAddress;
        v.voterName = _voterName;
        v.voted = false;
        voterRegistration[_voterAddress] = v;
        totalVoter++;
    }

    function startVote()
        public
        inState(State.Created)
        onlyOfficial
    {
        state = State.Voting;
    }

    function doVote(bool _choice)
        public
        inState(State.Voting)
        returns (bool voted)
    {
        bool found = false;
```

Figure 7 : Smart Contract 3

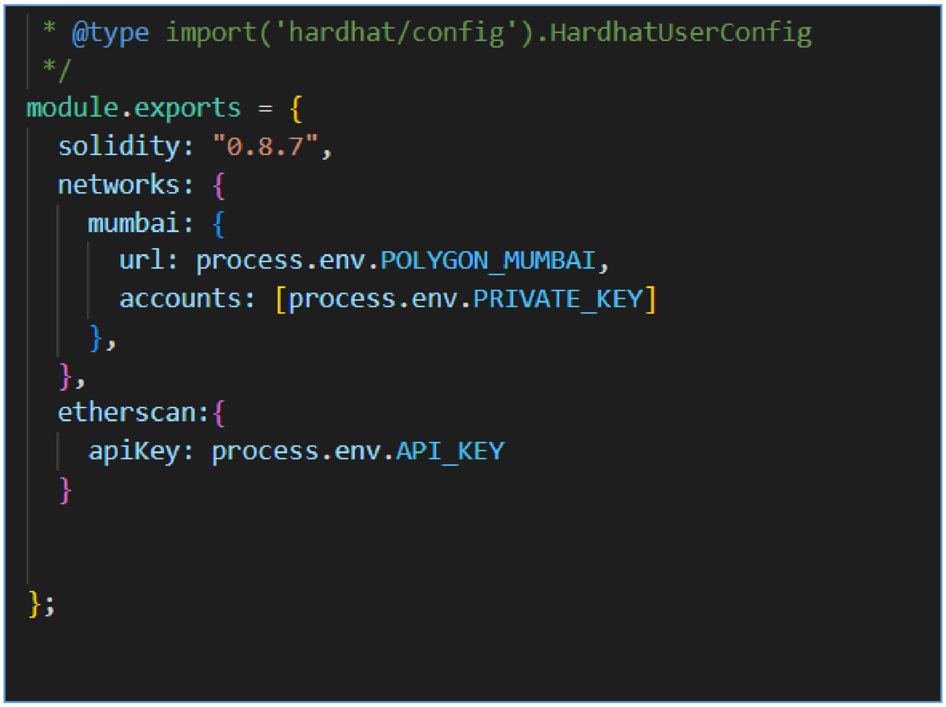


```

        if (bytes(voterRegistration[msg.sender].voterName).length != 0
        && !voterRegistration[msg.sender].voted){
            voterRegistration[msg.sender].voted =true;
            vote memory v;
            v.voterAddress =msg.sender;
            v.choice = _choice;

            if(_choice){
                countResult++;
            }
            votes[totalVote] =v;
            totalVote++;
            found = true;
        }
        return found;

    }

    function endVote()
        public
        inState(State.Voting)
        onlyOfficial
    {
        state = State.Ended;
        finalResult = countResult;

    }

}
```

Figure 8 : Smart Contract 4

Solidity was used to create the product's backbone, which is a Smart Contract. Solidity is an object-oriented high-level programming language designed specifically for smart contract creation and development. There are a variety of platforms for working with Solidity, but Remix is the most popular option since it includes its own block-chain for testing the contract while it is being built.

After building smart contract, it was deployed on the block-chain. For this, the test net Mumbai by PolygonScan was used.

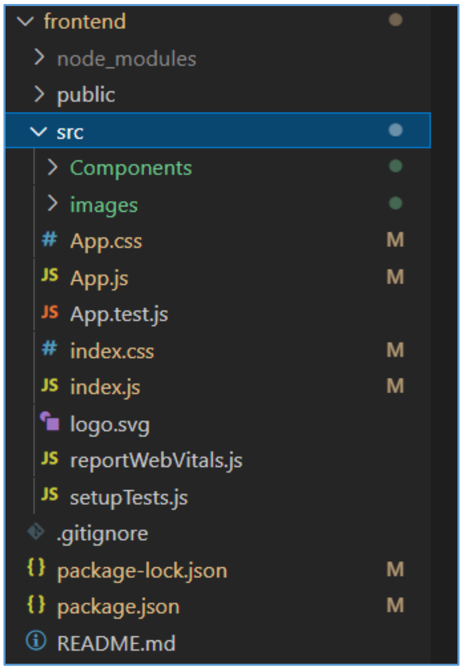


```
 * @type import('hardhat/config').HardhatUserConfig
 */
module.exports = {
  solidity: "0.8.7",
  networks: {
    mumbai: {
      url: process.env.POLYGON_MUMBAI,
      accounts: [process.env.PRIVATE_KEY]
    },
  },
  etherscan:{
    apiKey: process.env.API_KEY
  }


};
```

Figure 9 : HardHat Contract Deploy

Deploying Smart Contracts on the block-chain entails a cost that must be paid in crypto, hence the usage of a test network is utilized to solve this issue and make the procedure less expensive. Web3 engineers are familiar with test networks, and it's a good idea to test any contract on one before releasing it to the main network. Mumbai, which is powered by Polygon, serves as the product's test network.

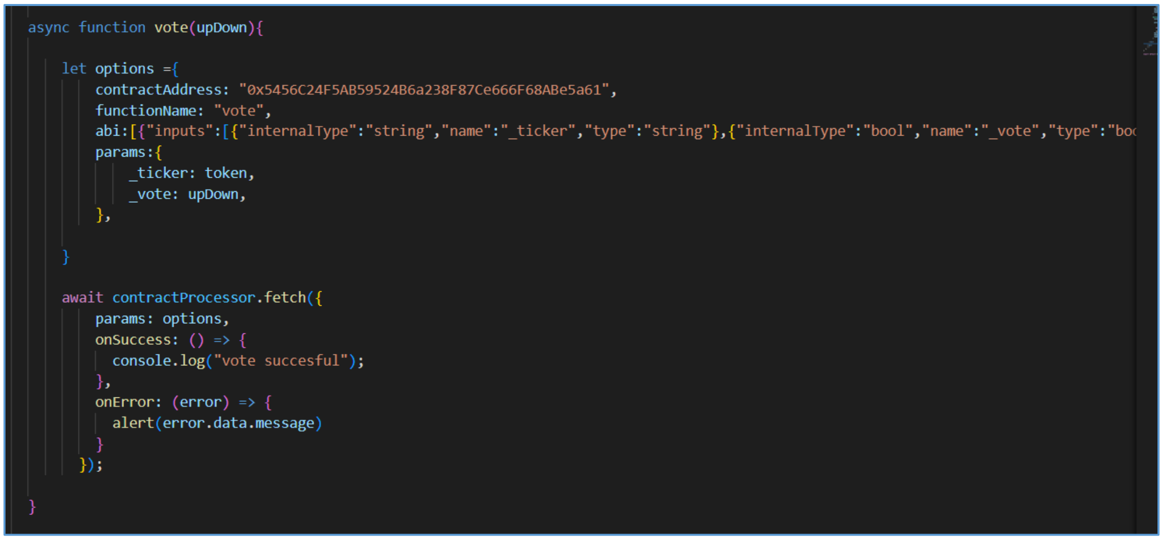
After contract deploy, The front end was designed using React App. React framework was used to accomplish the Front End



```
frontend
  node_modules
  public
  src
    Components
    images
    App.css                M
    App.js                 M
    App.test.js
    index.css              M
    index.js               M
    logo.svg
    reportWebVitals.js
    setupTests.js
  .gitignore
  package-lock.json        M
  package.json             M
  README.md
```

Figure 10 : React Frame Work

Finally, the mechanisms for linking the webpage with the smart contract, which were both developed earlier in the process, were discovered. This assignment was completed with the help of Moralis and its tools. Moralis also offered the server for extracting data directly from the block-chain. The front end was connected to the smart contract capability of voting on the various polls, and now the user may activate a block-chain transaction directly through our one page app.



```
async function vote(upDown){

    let options ={
        contractAddress: "0x5456C24F5AB59524B6a238F87Ce666F68ABe5a61",
        functionName: "vote",
        abi:[{"inputs":[{"internalType":"string","name":"_ticker","type":"string"},{"internalType":"bool","name":"_vote","type":"boo
        params:{
            _ticker: token,
            _vote: upDown,
        },

    }

    await contractProcessor.fetch({
        params: options,
        onSuccess: () => {
          console.log("vote succesful");
        },
        onError: (error) => {
          alert(error.data.message)
        }
      });

}
```

Figure 11 : Smart contract connection Code

ii. Testing

There are two ways for user to interact with the system. One is the through the front end app created using React and the other way is straight through the contract deployed on out Mumbai Test Net Network.

Firstly demonstrating interaction through the website.

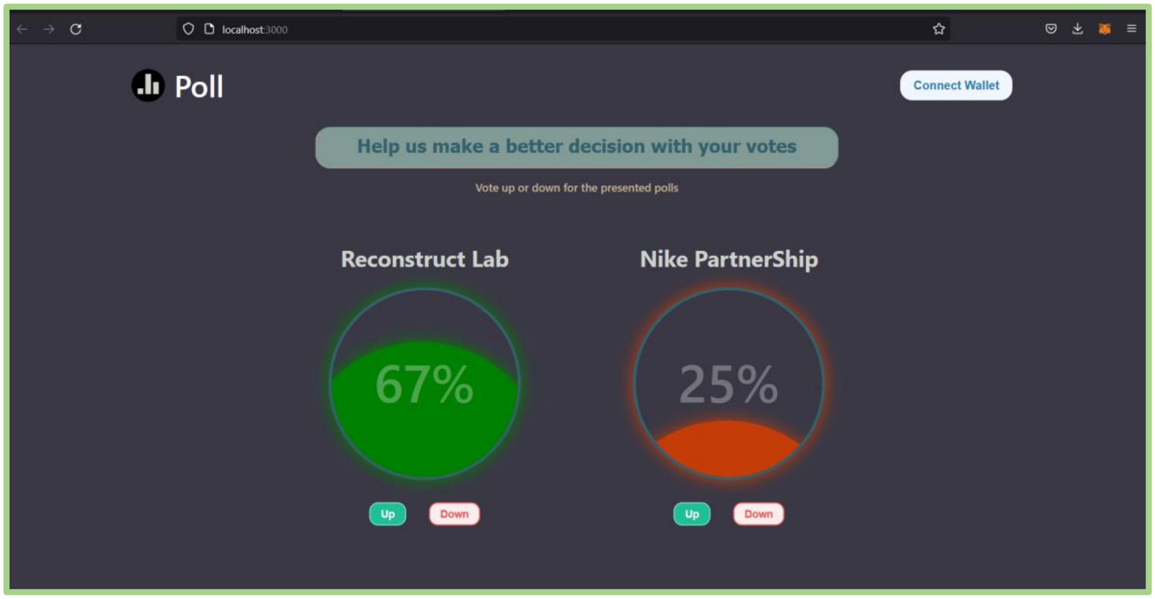

localhost:3000
Poll
Connect Wallet
Help us make a better decision with your votes
Vote up or down for the presented polls
Reconstruct Lab
67%
Up
Down
Nike PartnerShip
25%
Up
Down

Figure 12 : Home page of App

This is the Website that will be presented to the user. Here, the available events to vote on are shown to the user, where the user can make vote accordingly.

Step 1: Wallet Connection

Firstly, the user needs have a crypto wallet setup.

Connect your crypto wallet by clicking the “Connect Wallet” button on the top right corner of the webpage.

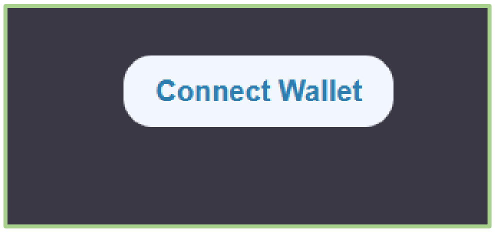


Figure 13 : Wallet connect button

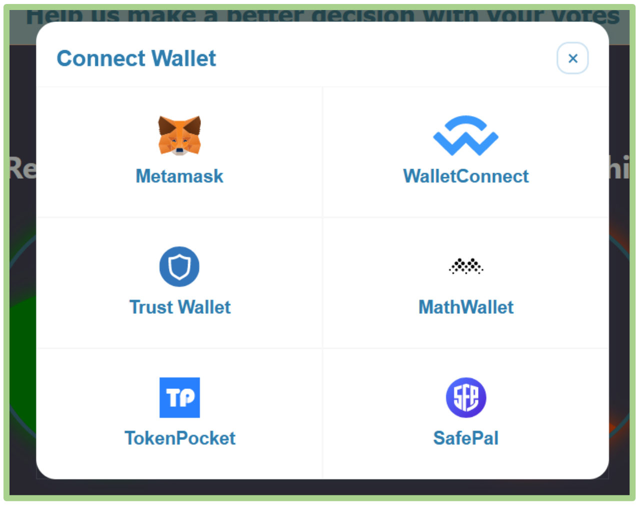
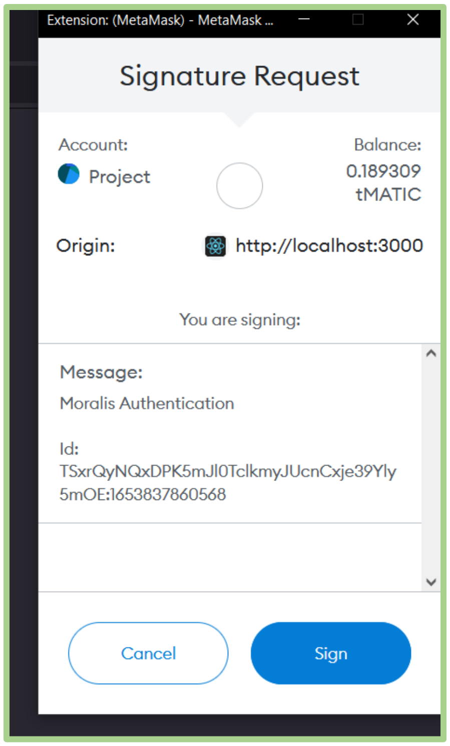


Figure 14 : Wallet choice page

Here, the user can select the desired wallet the user want to connect.



Signature Request

Account: Project

Balance: 0.189309 tMATIC

Origin: http://localhost:3000

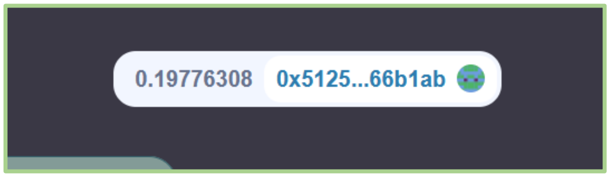
You are signing:

Message:
Moralis Authentication

Id:
TSxrQyNQxDPK5mJl0TclkmyJUcnCxje39Yly5mOE:1653837860568

Figure 15 : wallet connection sign request

The user will be prompted with a Signature Request on the wallet asking to authorize their wallet to connect with the site. Sign the message.



0.19776308 0x5125...66b1ab

Figure 16 : wallet information

This is how the top right corner should look after the connection.

Step 2: Vote on the available polls

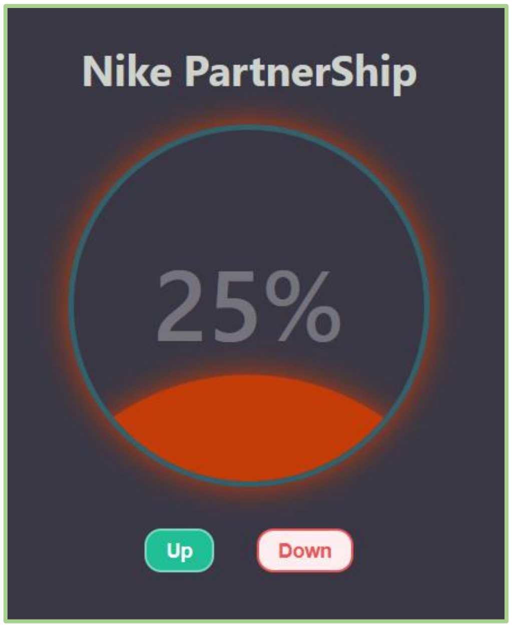
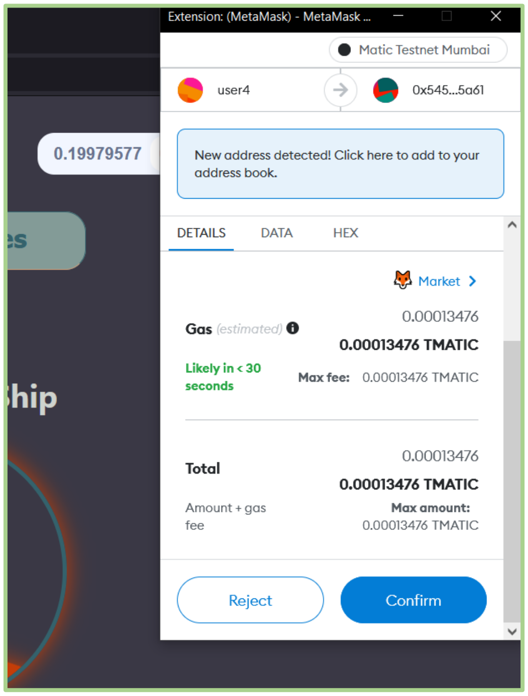


Figure 17 : Poll status before voting

The user can interact with the poll with these two up or down buttons visible.

Press up or down and the user will be prompted with a wallet transaction confirmation.



user4 → 0x545...5a61

New address detected! Click here to add to your address book.

Gas (estimated) 0.00013476
0.00013476 TMATIC
Likely in < 30 seconds
Max fee: 0.00013476 TMATIC

Total 0.00013476
0.00013476 TMATIC
Amount + gas fee
Max amount: 0.00013476 TMATIC

Figure 18 : Transaction popup for Vote

Confirm the transaction
Note: The user needs to have some TMATIC (Test-MATIC) in there wallet. Get free TMATIC from the website “https://faucet.polygon.technology/” in case of empty wallet.

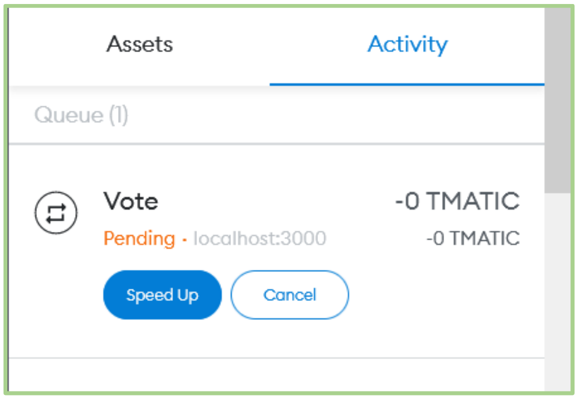


Figure 19 : Transaction Pending

After confirmation, the transaction should have a pending status on the wallet waiting for gas requirements met.

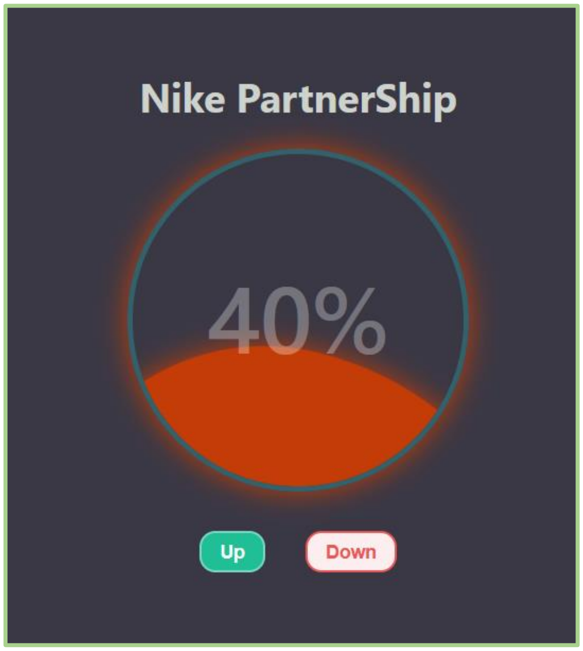
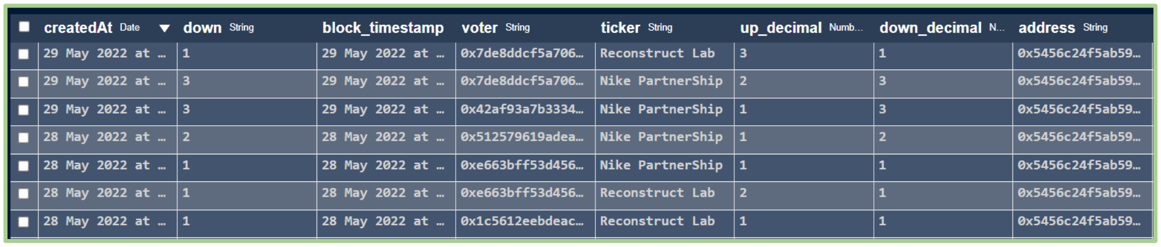


Figure 20 : Poll status after vote

After that the user can see live change in the poll results the user chose to vote on.



□	createdAt Date ▼	down String	block_timestamp	voter String	ticker String	up_decimal Numb...	down_decimal N...	address String
□	29 May 2022 at …	1	29 May 2022 at …	0x7de8ddcf5a706…	Reconstruct Lab	3	1	0x5456c24f5ab59…
□	29 May 2022 at …	3	29 May 2022 at …	0x7de8ddcf5a706…	Nike PartnerShip	2	3	0x5456c24f5ab59…
□	29 May 2022 at …	3	29 May 2022 at …	0x42af93a7b3334…	Nike PartnerShip	1	3	0x5456c24f5ab59…
□	28 May 2022 at …	2	28 May 2022 at …	0x512579619adea…	Nike PartnerShip	1	2	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0xe663bff53d456…	Nike PartnerShip	1	1	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0xe663bff53d456…	Reconstruct Lab	2	1	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0x1c5612eebdeac…	Reconstruct Lab	1	1	0x5456c24f5ab59…

Figure 21 : Database showing new votes

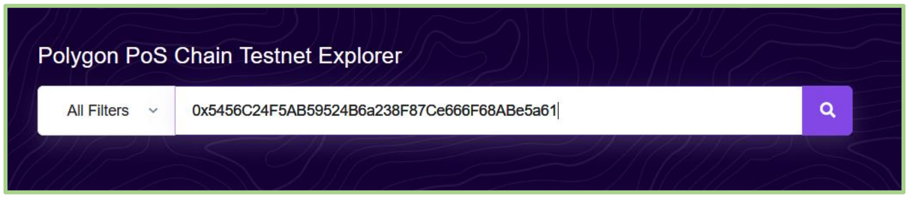
The user has successfully voted using the block-chain polling system. The database shows new entries and thus the voting through the front end is a success.

Now, demonstrating the user interaction thorough the smart contract.

Firstly, go on the Mumbai polygon block-chain website

https://mumbai.polygonscan.com/

Then search for the polls contract address 0x5456C24F5AB59524B6a238F87Ce666F68ABe5a61

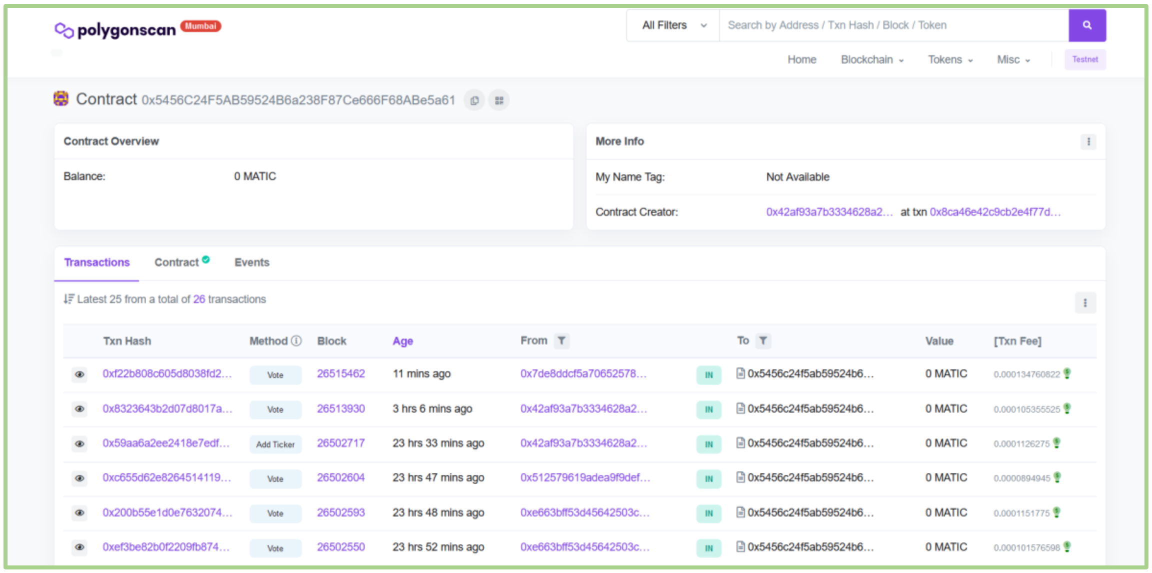


Polygon PoS Chain Testnet Explorer

0x5456C24F5AB59524B6a238F87Ce666F68ABe5a61

Figure 22 : Searching Wallet address on PolygonScan

Put in the address of the polling contract and press search.



Contract 0x5456C24F5AB59524B6a238F87Ce666F68ABe5a61

Contract Overview

Balance: 0 MATIC

More Info

My Name Tag: Not Available

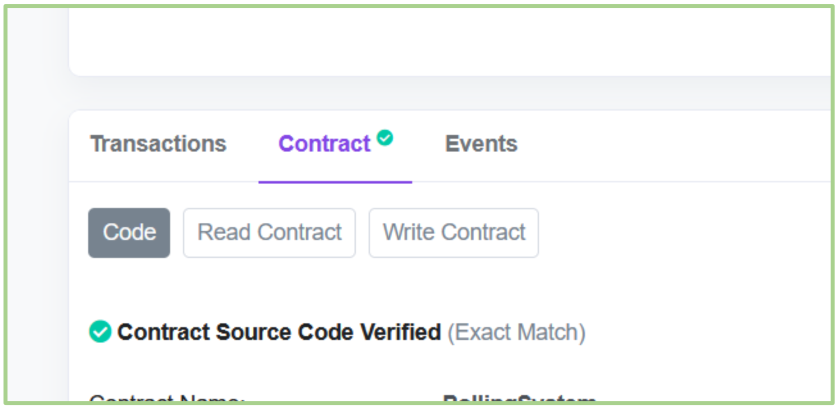
Contract Creator: 0x42af93a7b3334628a2… at txn 0x8ca46e42c9cb2e4f77d…

Latest 25 from a total of 26 transactions

Txn Hash	Method	Block	Age	From		To	Value	[Txn Fee]
0xf22b808c605d8038fd2…	Vote	26515462	11 mins ago	0x7de8ddcf5a70652578…	IN	0x5456c24f5ab59524b6…	0 MATIC	0.000134760822
0x8323643b2d07d8017a…	Vote	26513930	3 hrs 6 mins ago	0x42af93a7b3334628a2…	IN	0x5456c24f5ab59524b6…	0 MATIC	0.000105355525
0x59aa6a2ee2418e7edf…	Add Ticker	26502717	23 hrs 33 mins ago	0x42af93a7b3334628a2…	IN	0x5456c24f5ab59524b6…	0 MATIC	0.0001126275
0xc655d62e8264514119…	Vote	26502604	23 hrs 47 mins ago	0x512579619adea9f9def…	IN	0x5456c24f5ab59524b6…	0 MATIC	0.0000894945
0x200b55e1d0e7632074…	Vote	26502593	23 hrs 48 mins ago	0xe663bff53d45642503c…	IN	0x5456c24f5ab59524b6…	0 MATIC	0.0001151775
0xef3be82b0f2209fb874…	Vote	26502550	23 hrs 52 mins ago	0xe663bff53d45642503c…	IN	0x5456c24f5ab59524b6…	0 MATIC	0.000101576598

Figure 23 : Contract View on PolygonScan

The user will be presented with this screen



Contract Source Code Verified (Exact Match)

Contract Name: RollingSystem

Figure 24 : Contract page

Press on contract

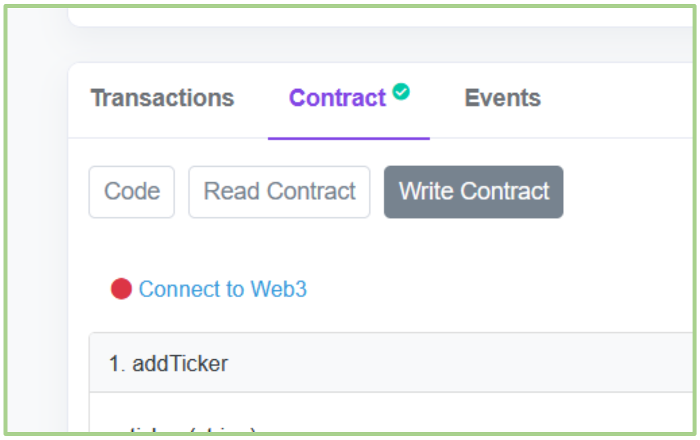
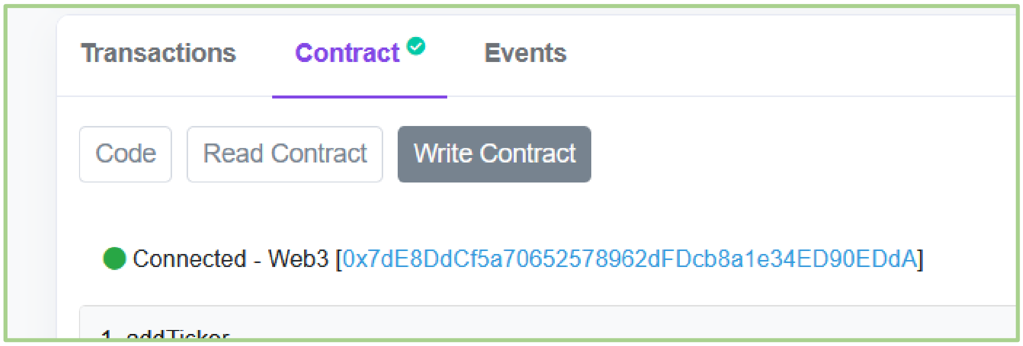


Figure 25 : Write tab on Contract

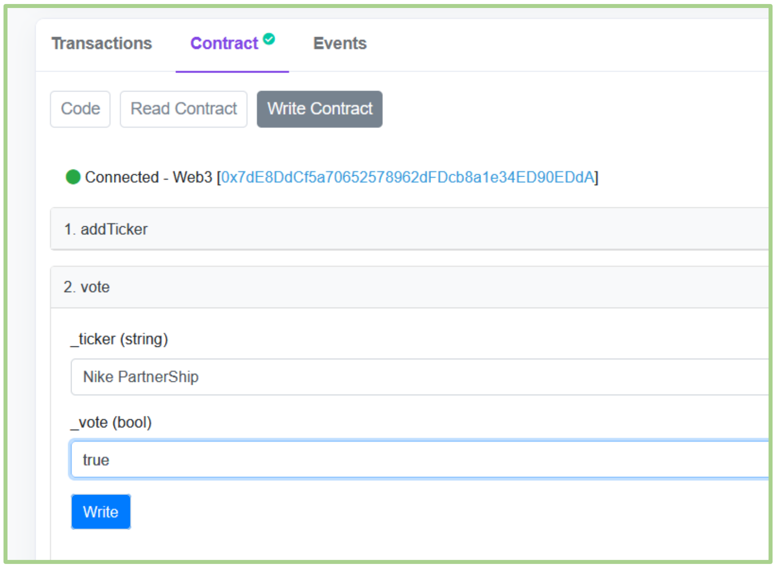
And then on Write Contract



● Connected - Web3 [0x7dE8DdCf5a70652578962dFDcb8a1e34ED90EDdA]

Figure 26 : Wallet connect with smart Contract

Press on the “Connect to Web3” and connect your wallet.



● Connected - Web3 [0x7dE8DdCf5a70652578962dFDcb8a1e34ED90EDdA]

1. addTicker

2. vote

_ticker (string)

Nike PartnerShip

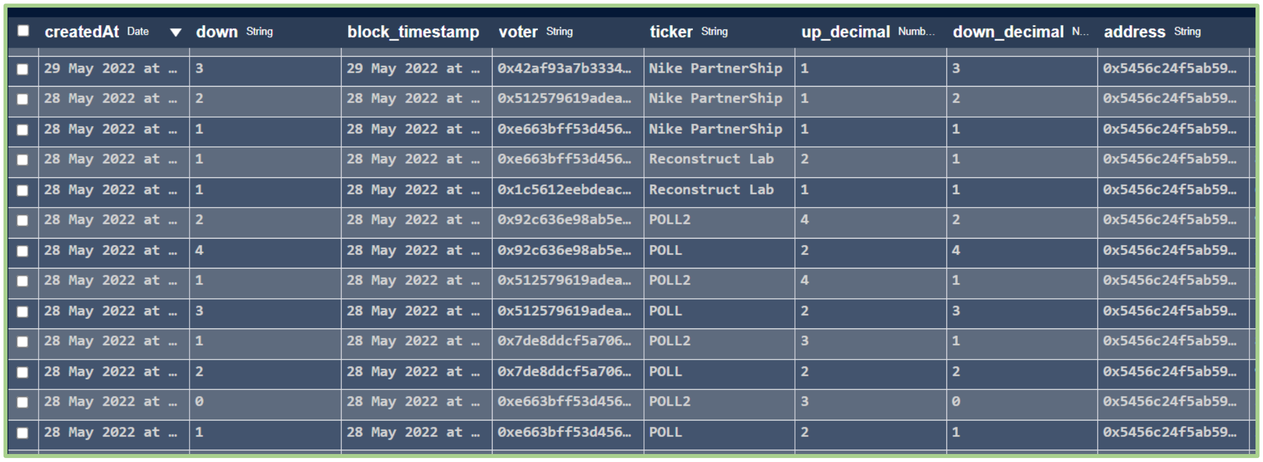
_vote (bool)

true

Write

Figure 27 : Writing vote on smart contract

Provide the name of the poll in the Vote section and provide a Boolean value (true or false) and press "Write" This should prompt a transaction on your wallet and confirming it will be the last step.



□	createdAt Date ▼	down String	block_timestamp	voter String	ticker String	up_decimal Numb…	down_decimal N…	address String
□	29 May 2022 at …	3	29 May 2022 at …	0x42af93a7b3334…	Nike PartnerShip	1	3	0x5456c24f5ab59…
□	28 May 2022 at …	2	28 May 2022 at …	0x512579619adea…	Nike PartnerShip	1	2	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0xe663bff53d456…	Nike PartnerShip	1	1	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0xe663bff53d456…	Reconstruct Lab	2	1	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0x1c5612eebdeac…	Reconstruct Lab	1	1	0x5456c24f5ab59…
□	28 May 2022 at …	2	28 May 2022 at …	0x92c636e98ab5e…	POLL2	4	2	0x5456c24f5ab59…
□	28 May 2022 at …	4	28 May 2022 at …	0x92c636e98ab5e…	POLL	2	4	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0x512579619adea…	POLL2	4	1	0x5456c24f5ab59…
□	28 May 2022 at …	3	28 May 2022 at …	0x512579619adea…	POLL	2	3	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0x7de8ddcf5a706…	POLL2	3	1	0x5456c24f5ab59…
□	28 May 2022 at …	2	28 May 2022 at …	0x7de8ddcf5a706…	POLL	2	2	0x5456c24f5ab59…
□	28 May 2022 at …	0	28 May 2022 at …	0xe663bff53d456…	POLL2	3	0	0x5456c24f5ab59…
□	28 May 2022 at …	1	28 May 2022 at …	0xe663bff53d456…	POLL	2	1	0x5456c24f5ab59…

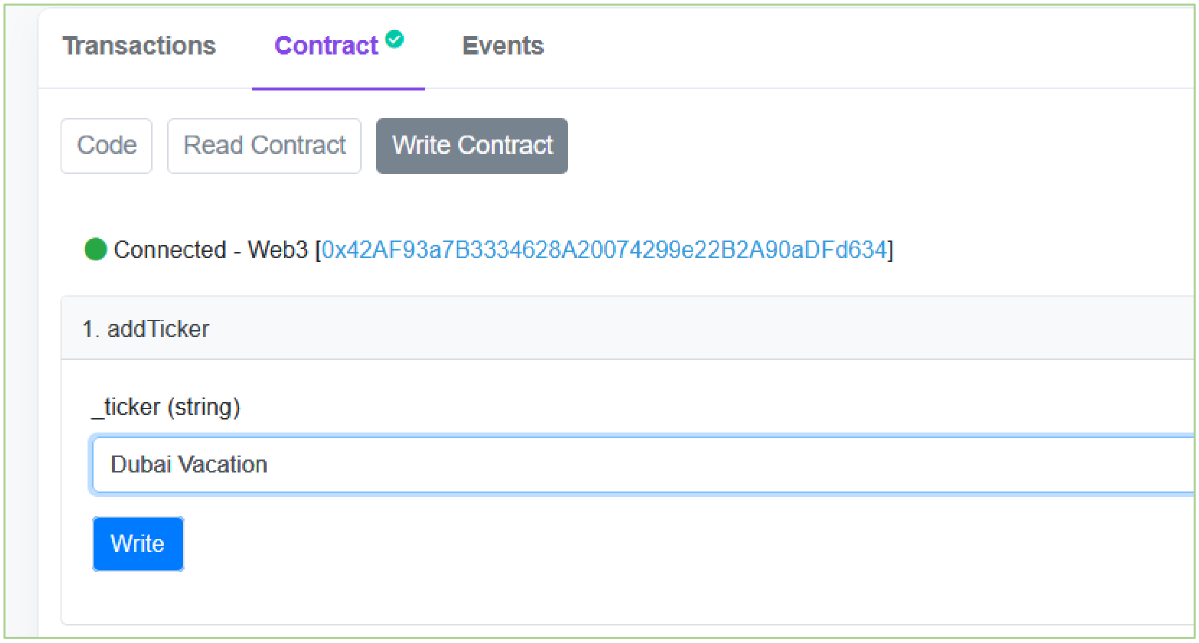
Figure 28 : Database with entries thought contract

Here, the user interaction went smoothly as expected thus completing the testing from the user side with a success.

Admin Special Interaction: Adding Polls

Also, the Admin (The contract creator) can add new polls using the Smart Contract.

Connect with the contract using the Admins wallet,



● Connected - Web3 [0x42AF93a7B3334628A20074299e22B2A90aDFd634]

1. addTicker

_ticker (string)

Dubai Vacation

Figure 29 : Writing on Contract

Writing in the addTicker box, the admin can add new polls with desired name, for which the user can later vote on.

This concludes the testing for our product. It can be seen that the product works just fine and is able to take in votes from different users and is able to show updated results of the vote live on the webpage.

8. Product Evaluation

The product focus on the making a polling system focused around smarts contracts and web interaction all utilizing the block-chain technology making the perfect D-app for a voting system. Its simple contract, that focuses on making a simple new poll additions and making it interactable with votes, makes the system very easy to read and be understood by anyone even with simple programming knowledge. These simple tasks of voting and adding new polls is all handled by the contract directly, this introduces a whole new way of approaching programming and providing utility. All votes being made are going straight through the contract to get validated providing security against data manipulation and rigged results. As contract is all it needs for actual working system, the system becomes easier to audit and this helps by saving extra time and recourses of the company. Thought contract is the backbone of the structure, it still lacks an interactive platform that is valid for the modern IT advancement. This is where the front end of the app, our react based webpage, comes in handy. This provides the user a better experience for interacting with the contract. The webpage is directly linked with the smart contract making use of the very same smart contract but in a more refined and simpler method. This changes nothing in the actual functionality of the D-app and adds an easier user-friendly method. A user-friendly app focuses on the user end more and aims to deliver a smooth experience to the users in the motive of generating more interactions and collection of better data. The app successfully makes the user experience very simplified though its simple webpage where user only has to connect the wallet and approve the transaction for voting on the selected poll with desired agreement. Also, the admin can add new polls to the contract database for future polls needed. This provides sustainability to the product where it can be used as a default tool for conducting polls in a company.

The smart contract is written in a way where it can detect any proxy multiple votes being made by the same user through the use of simple conditions. This is done by limiting the voting feature to one vote for one wallet. This is very mandatory as any multiple votes on a single poll makes the results of the poll inaccurate defeating the purpose of authenticity and block-chain database functionality. The site is connect with smart contract with the help of web3 technology and makes use of database where data is pulled out directly from the block. The database provided by Moralis allows the app developers pull the transactions record made on the block-chain and makes it view-able in a more informative way. The use of this database is very helpful as the data it holds represents the total number of votes being made on different polls and events. The app has a built graphical non interactive result displayer for the different polls. This creates a new functionality of company's member sentiment towards the company and directly provides data on active poll for the user only working with the front end.

As all product needs time and updates to reach the ideal stage of its final form, the product also has few missing attributes that can be a possible and ideally a wise upgrade for the future of the product. Firstly the major missing attribute is the process of adding new votes on the website by admin is not automatic and involves with code interferences. Thought he admin can add new polls on the smart contract and user can vote on them through the smart contract, for the user to be able to do so though web page is not yet possible. Also, other missing feature would be a connection so that the polling system can be integrated with the company and can be easily accessible for the company members.

Evaluating the product is a tabular form for easier understanding:

Features	Expectations	Results & Evaluation
Smart Contract	Deployable on the block-chain	Passed
Web page	Have a intractable web-page	Passed

Adding votes on poll	User should be able to add votes on poll	Passes
Show vote results	Users should be able to see vote results	Passed
Add new poll	Admin should be able to add new polls	Passed
Voting Eligibility	Users should not be able to vote twice on the same poll	Passed
Wallet	Users cannot interact with the page without a wallet connection	Passed
Transparent System	The system should be transparent	Passed

Figure 30 : Product Evaulation with Expectation and Result

9. Project Evaluation

The product was created using a variety of resources, including web2 and web3 components, mostly web3. Poll is a block-chain-based polling/voting tool that combines web2 functionality with web3 authenticity. It uses the web2 website platform, but connects to a smart contract that works as the backbone for all transaction votes, keeping all data secure and public and unconnected to any particular organization. This solidity-based smart contract focuses on creating new poll commands as well as voting on them. This program has a very broad scope and can be used by a variety of parties and organizations. The project is supposed for a small business to utilize in order to make democratic strategic choices based on input from colleagues and company team members.

This will enable the company to make active decisions about renovations and future goals for the company, as well as actively work toward its improvement. When it comes to online voting, the confidence that block-chain and smart contracts provide is the highest, which increases the accuracy and interest in the decision to be made. The voting process has also been streamlined, which has resulted in an increase in active voting members, allowing the company to achieve higher numbers and gather more information. The product is created in such a manner that comprehending the app is simple and requires little effort on the part of the user. The user is presented with a single page app that displays the existing polls and includes two buttons for voting up or down according to the user's preference. When a decision is taken and the button is pressed, the smart contract is triggered, which checks the authenticity of the user address that is now associated with the page. If the user has already cast a vote in the poll, the smart contract will not allow any more transactions to take place.

Every phase was finished before moving on to the next, and further research was conducted in each phase to ensure a thorough grasp of the end product as well as the possibilities these components provide for the product's future. The Waterfall Methodology was used to finish the procedure one step at a time. As predicted, the procedure had to deal with some of the waterfall method's key flaws. Due to compatibility concerns subsequently discovered when the product was being created in the front end phases, the smart contract launched had to be deployed using another application. Similarly, when linking the smart contract to the frontend, adjustments to the front end were required, which was both inconvenient and time consuming.

Solidity was used to create the product's backbone, which is a Smart Contract. Solidity is an object-oriented high-level programming language designed specifically for smart contract creation and development. There are a variety of platforms for working with Solidity, but Remix is the most popular option since it includes its own block-chain for testing the contract while it is being built. This critical feature enables the programmer to verify the smart contract's operation and guarantee that the final contact is fully audited before it is implemented in the main block-chain. With the aid of React, the front end was built. React is a JavaScript library that focuses on front-end development. (Platforms, n.d.) Reacts distinctive feature is its ability to create single-page user interfaces in which just the necessary elements of the webpage are altered when the user interacts, maintaining the entire page as one without the need to request a new page load from the server. This results in a smoother and more comfortable user interaction, as well as a better overall user experience. On the front end, React and CSS components were used to create the overall design. CSS (Cascading Style Sheets) is a style sheet language that focuses on the styling of a webpage, making it a must-have for the front end. The product is constructed in such a way that it takes little user input and displays the results of the voting in a graphical format. Created polls will be shown on the app's top page, along with the poll's current score and two easy buttons for voting up or down. Interacting with the buttons will initiate a transaction, which will result in a visible change in the poll result.

Overall, Detailed research on Block-chain technology and smart contract as well as web3 resources was needed for the completion of the project. The targeted features for the product was successfully met and some additional future upgrades were planned.

10. Summary and Conclusion

A polling system is a voting mechanism that uses block-chain technology to demonstrate the block-and chain's smart contract's transparency. In a business, decisions must be taken and this product focuses on the sentiment of the polls being conducted, allowing users to cast their votes and viewing the live results on the homepage, which indicate the sentiment of the total votes cast in favor or against. Block-chain technology has been widely accepted in the new decade, and many major organizations have begun to implement it. The breadth and functionality of smart contracts and voting systems are tested on the Block-chain in this project. When it comes to online voting, the confidence that block-chain and smart contracts give is the highest, which increases the accuracy and interest in the choice to be made. The voting procedure has also been streamlined, which has resulted in an increase in active voting members, allowing the firm to achieve higher numbers and gather more data.

A Smart Contract is a collection of programmable codes that encodes a system's structure for transaction validation. It has a collection of rules and hard-coded assertions that, when followed, will only enable transactions between parties to take place. The smart Contract was created and tested utilizing both the Remix local block-chain environment and the Hardhat local block-chain. Then the smart contract was deployed on the test net provided by Mumbai for further implementation with front end. Front end was created using React App environment and successfully connected with smart contract using Moralis Tools.

Before going on to the next step, each phase was completed, and more research was undertaken in each to ensure a full understanding of the end product as well as the possibilities these components present for the product's future. The method was completed one step at a time using the Waterfall Methodology. The Waterfall Methodology has shown to be effective throughout the development process. If the product were to be given to a team, other methodologies, particularly Agile Methodology, would be beneficial.

The software is designed in such a way that understanding it is straightforward and needs minimal effort on the user's behalf. The user is provided with a single-page app that displays the current polls and contains two buttons for voting up or down based on the user's preferences. The smart contract is constructed in such a way that simple criteria may be used to identify any proxy multiple votes by the same user. This is accomplished by restricting the voting capability to to one vote per wallet. This is extremely important since numerous votes on a single poll skews the results, negating the purpose of authenticity and block-chain database functioning. The site uses web3 technology to link to smart contracts and makes use of a database that pulls data straight from the block. The database supplied by Moralis allows app developers to retrieve the block-chain transaction records and display them in a more meaningful manner. This database is extremely useful since the information it contains indicates the overall number of votes cast in various polls and events. For each poll, the app contains a built-in graphical non-interactive result displayer. This adds a new feature that shows how a company's members feel about it and gives statistics on a current poll for users who just interact with the front end. The product contains a few lacking qualities that may be a viable and preferably a prudent upgrade for the future of the product, since all products require time and updates to reach the optimum stage of their ultimate form.

Block-chain technology has made it possible for this product to be created in the first place. This truly magnificent technology has changed how things were adopted on the internet. The truly decentralized system can be achievable with the help of Block-chain technology. Smart contracts are the rule book for the products developed with block-chain technology. By the help of which, plus other web2 and web3 resources, the polling system was successfully created and is fully functional for use.

• Bibliography

Anon., n.d. *What are smart contracts on blockchain?.* [Online] Available at: https://www.ibm.com/topics/smart-contracts
[Accessed 30 05 2022].

Brush, K., n.d. *Tech Target.* [Online] Available at: https://www.techtarget.com/searchsoftwarequality/definition/use-case
[Accessed 12 06 2022].

Iansiti, M. & Karim, R. L., 2017. *Harvard Business Review.* [Online] Available at: https://hbr.org/2017/01/the-truth-about-blockchain#:~:text=Blockchain%E2%80%94a%20peer%2Dto%2D,transferring%20ownership%2C%20and%20confirming%20transactions.
[Accessed 30 5 2022].

Little, D., 2014. *What is methodology?.* [Online] Available at: https://understandingsociety.blogspot.com/2014/08/what-is-methodology.html
[Accessed 31 5 2022].

Platforms, M., n.d. *React.* [Online] Available at: https://reactjs.org/
[Accessed 30 5 2022].

Senay, D. & Resul , D., 2018. *Software requirement analysis: Research challenges and technical approaches.* Istanbul, s.n.

Shafaq , N. K. et al., 2021. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer networking and applications,* 14(2901-2925), p. 5.

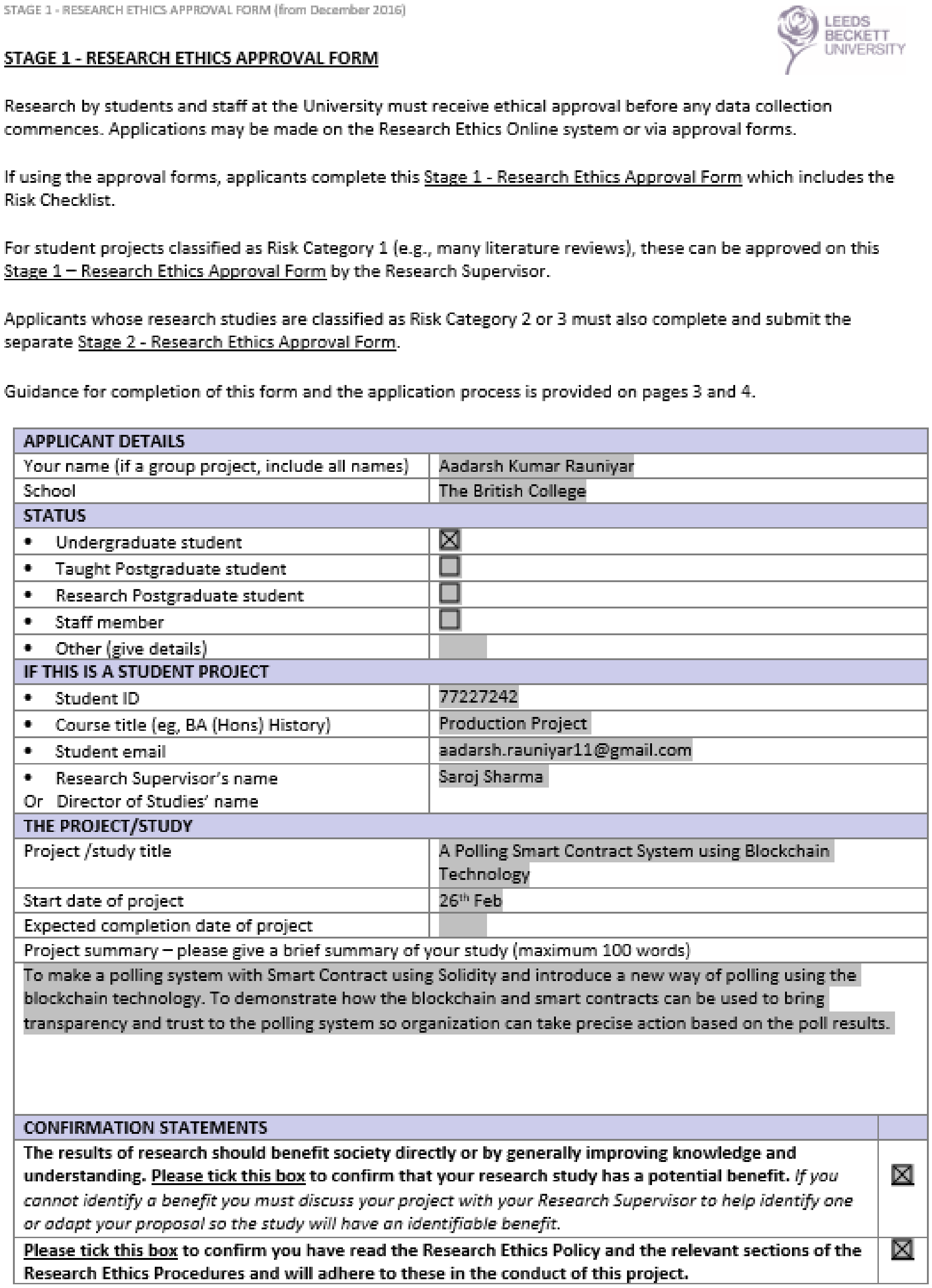
Shaku , L. A., Hung, K. P. & Joa, S. L., 2019. Abstract. *Development of a Distributed Blockchain eVoting System.*

Software, 5., 2020. *Top 5 Methodologies Used In Mobile App Development.* [Online]
Available at: https://www.5280software.net/blog-post/top-5-methodologies-used-in-mobile-app-development/
[Accessed 31 5 2022].

Weldemariam, K., 2010. A Survey: Electronic Voting Development and Trends..

Appendices

Ethical Consent Form



STAGE 1 - RESEARCH ETHICS APPROVAL FORM (from December 2016)

LEEDS BECKETT UNIVERSITY

STAGE 1 - RESEARCH ETHICS APPROVAL FORM

Research by students and staff at the University must receive ethical approval before any data collection commences. Applications may be made on the Research Ethics Online system or via approval forms.

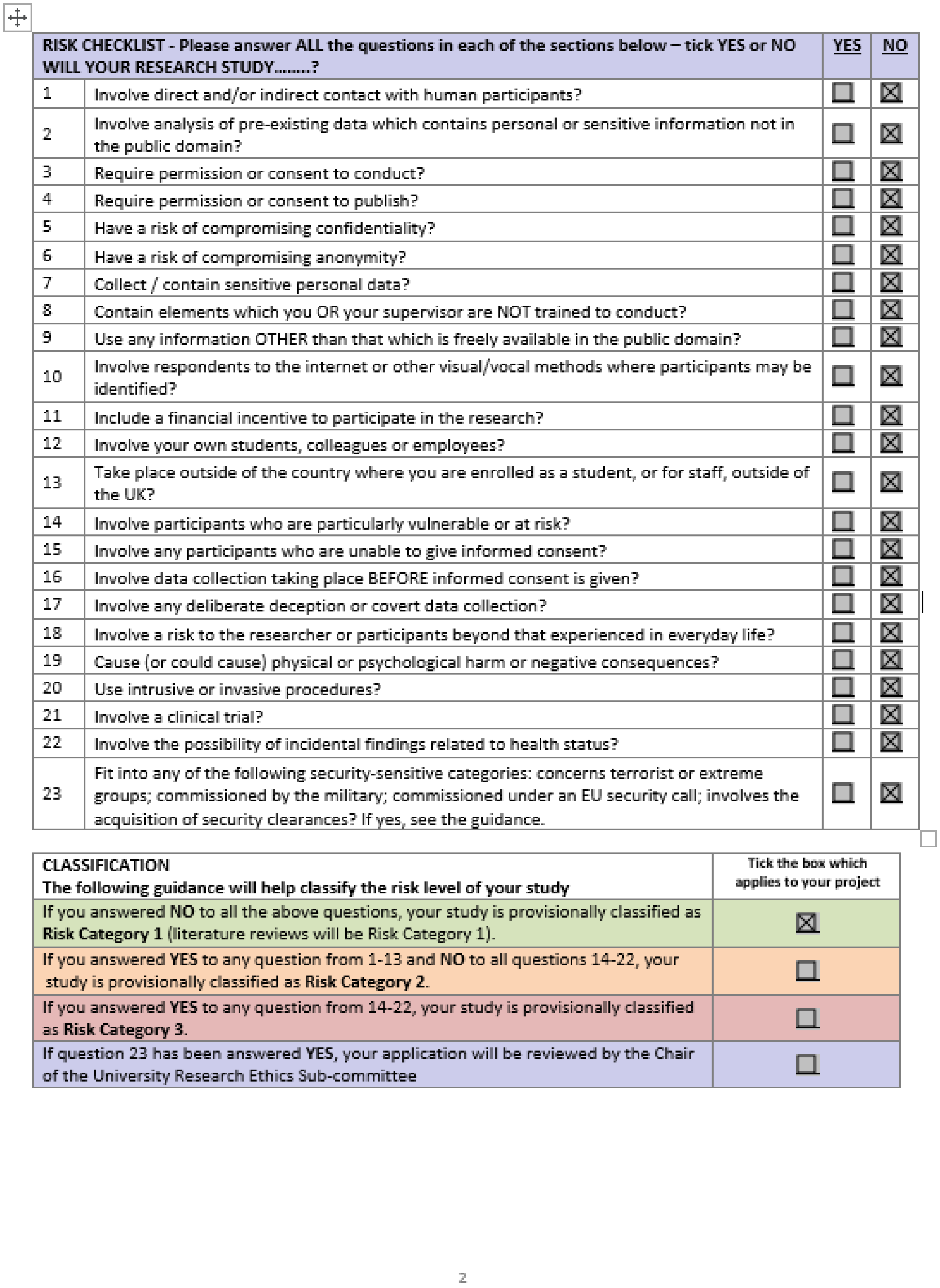
If using the approval forms, applicants complete this Stage 1 - Research Ethics Approval Form which includes the Risk Checklist.

For student projects classified as Risk Category 1 (e.g., many literature reviews), these can be approved on this Stage 1 – Research Ethics Approval Form by the Research Supervisor.

Applicants whose research studies are classified as Risk Category 2 or 3 must also complete and submit the separate Stage 2 - Research Ethics Approval Form.

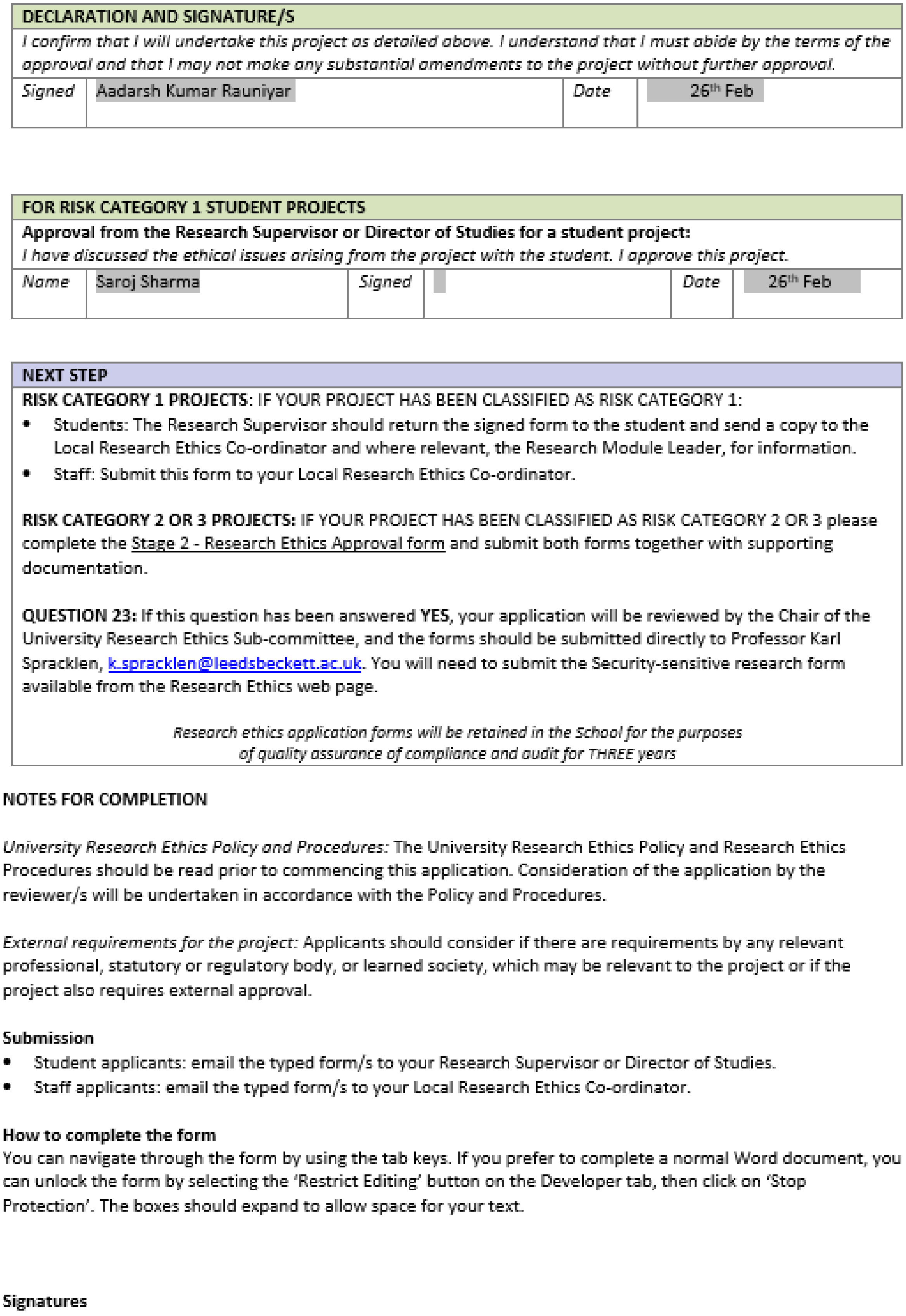
Guidance for completion of this form and the application process is provided on pages 3 and 4.

APPLICANT DETAILS	
Your name (if a group project, include all names)	Aadarsh Kumar Rauniyar
School	The British College
STATUS	
• Undergraduate student	☒
• Taught Postgraduate student	☐
• Research Postgraduate student	☐
• Staff member	☐
• Other (give details)	
IF THIS IS A STUDENT PROJECT	
• Student ID	77227242
• Course title (eg, BA (Hons) History)	Production Project
• Student email	aadarsh.rauniyar11@gmail.com
• Research Supervisor's name Or Director of Studies' name	Saroj Sharma
THE PROJECT/STUDY	
Project /study title	A Polling Smart Contract System using Blockchain Technology
Start date of project	26th Feb
Expected completion date of project	
Project summary – please give a brief summary of your study (maximum 100 words)	
To make a polling system with Smart Contract using Solidity and introduce a new way of polling using the blockchain technology. To demonstrate how the blockchain and smart contracts can be used to bring transparency and trust to the polling system so organization can take precise action based on the poll results.	
CONFIRMATION STATEMENTS	
The results of research should benefit society directly or by generally improving knowledge and understanding. Please tick this box to confirm that your research study has a potential benefit. *If you cannot identify a benefit you must discuss your project with your Research Supervisor to help identify one or adapt your proposal so the study will have an identifiable benefit.*	☒
Please tick this box to confirm you have read the Research Ethics Policy and the relevant sections of the Research Ethics Procedures and will adhere to these in the conduct of this project.	☒



RISK CHECKLIST - Please answer ALL the questions in each of the sections below – tick YES or NO WILL YOUR RESEARCH STUDY……..?		YES	NO
1	Involve direct and/or indirect contact with human participants?	□	☒
2	Involve analysis of pre-existing data which contains personal or sensitive information not in the public domain?	□	☒
3	Require permission or consent to conduct?	□	☒
4	Require permission or consent to publish?	□	☒
5	Have a risk of compromising confidentiality?	□	☒
6	Have a risk of compromising anonymity?	□	☒
7	Collect / contain sensitive personal data?	□	☒
8	Contain elements which you OR your supervisor are NOT trained to conduct?	□	☒
9	Use any information OTHER than that which is freely available in the public domain?	□	☒
10	Involve respondents to the internet or other visual/vocal methods where participants may be identified?	□	☒
11	Include a financial incentive to participate in the research?	□	☒
12	Involve your own students, colleagues or employees?	□	☒
13	Take place outside of the country where you are enrolled as a student, or for staff, outside of the UK?	□	☒
14	Involve participants who are particularly vulnerable or at risk?	□	☒
15	Involve any participants who are unable to give informed consent?	□	☒
16	Involve data collection taking place BEFORE informed consent is given?	□	☒
17	Involve any deliberate deception or covert data collection?	□	☒
18	Involve a risk to the researcher or participants beyond that experienced in everyday life?	□	☒
19	Cause (or could cause) physical or psychological harm or negative consequences?	□	☒
20	Use intrusive or invasive procedures?	□	☒
21	Involve a clinical trial?	□	☒
22	Involve the possibility of incidental findings related to health status?	□	☒
23	Fit into any of the following security-sensitive categories: concerns terrorist or extreme groups; commissioned by the military; commissioned under an EU security call; involves the acquisition of security clearances? If yes, see the guidance.	□	☒

CLASSIFICATION **The following guidance will help classify the risk level of your study**	**Tick the box which applies to your project**
If you answered **NO** to all the above questions, your study is provisionally classified as **Risk Category 1** (literature reviews will be Risk Category 1).	☒
If you answered **YES** to any question from 1-13 and **NO** to all questions 14-22, your study is provisionally classified as **Risk Category 2**.	□
If you answered **YES** to any question from 14-22, your study is provisionally classified as **Risk Category 3**.	□
If question 23 has been answered **YES**, your application will be reviewed by the Chair of the University Research Ethics Sub-committee	□



DECLARATION AND SIGNATURE/S			
I confirm that I will undertake this project as detailed above. I understand that I must abide by the terms of the approval and that I may not make any substantial amendments to the project without further approval.			
Signed	Aadarsh Kumar Rauniyar	*Date*	26th Feb

FOR RISK CATEGORY 1 STUDENT PROJECTS					
Approval from the Research Supervisor or Director of Studies for a student project: *I have discussed the ethical issues arising from the project with the student. I approve this project.*					
Name	Saroj Sharma	*Signed*		*Date*	26th Feb

NEXT STEP
RISK CATEGORY 1 PROJECTS: IF YOUR PROJECT HAS BEEN CLASSIFIED AS RISK CATEGORY 1: • Students: The Research Supervisor should return the signed form to the student and send a copy to the Local Research Ethics Co-ordinator and where relevant, the Research Module Leader, for information. • Staff: Submit this form to your Local Research Ethics Co-ordinator. **RISK CATEGORY 2 OR 3 PROJECTS:** IF YOUR PROJECT HAS BEEN CLASSIFIED AS RISK CATEGORY 2 OR 3 please complete the Stage 2 - Research Ethics Approval form and submit both forms together with supporting documentation. **QUESTION 23:** If this question has been answered **YES**, your application will be reviewed by the Chair of the University Research Ethics Sub-committee, and the forms should be submitted directly to Professor Karl Spracklen, k.spracklen@leedsbeckett.ac.uk. You will need to submit the Security-sensitive research form available from the Research Ethics web page. *Research ethics application forms will be retained in the School for the purposes of quality assurance of compliance and audit for THREE years*

NOTES FOR COMPLETION

University Research Ethics Policy and Procedures: The University Research Ethics Policy and Research Ethics Procedures should be read prior to commencing this application. Consideration of the application by the reviewer/s will be undertaken in accordance with the Policy and Procedures.

External requirements for the project: Applicants should consider if there are requirements by any relevant professional, statutory or regulatory body, or learned society, which may be relevant to the project or if the project also requires external approval.

Submission

- Student applicants: email the typed form/s to your Research Supervisor or Director of Studies.
- Staff applicants: email the typed form/s to your Local Research Ethics Co-ordinator.

How to complete the form

You can navigate through the form by using the tab keys. If you prefer to complete a normal Word document, you can unlock the form by selecting the 'Restrict Editing' button on the Developer tab, then click on 'Stop Protection'. The boxes should expand to allow space for your text.

Signatures

Electronic/typed signatures are acceptable for emailed forms, as the emails provide the audit trail for all parties' agreement and approval of the forms (e.g., student applicant →Research Supervisor →Local Research Ethics Co-ordinator).

Outcome

Applicants will be advised of the outcome of the application by receipt of the signed form fromm:

- The Research Supervisor or Director of Studies for Risk Category 1 student projects;
- The Local Research Ethics Co-ordinator or the School level group for Risk Category 2 and 3 projects.

YOU MAY ONLY BEGIN ANY DATA COLLECTION ONCE YOU RECEIVE NOTIFICATION THAT THE PROJECT HAS ETHICAL APPROVAL. If the circumstances of your research study change after approval it is your responsibility to revisit the Risk Checklist and complete a further application.

Advice

When completing the Stage 1 - Research Ethics Approval Form,. if you are uncertain about the answer to any question, read the relevant section of the Research Ethics Procedures document, and if you are still unsure:

- if you are student, seek guidance from your Research Supervisor or Director of Studies;
- if you are a staff member, contact your Local Research Ethics Co-ordinator.

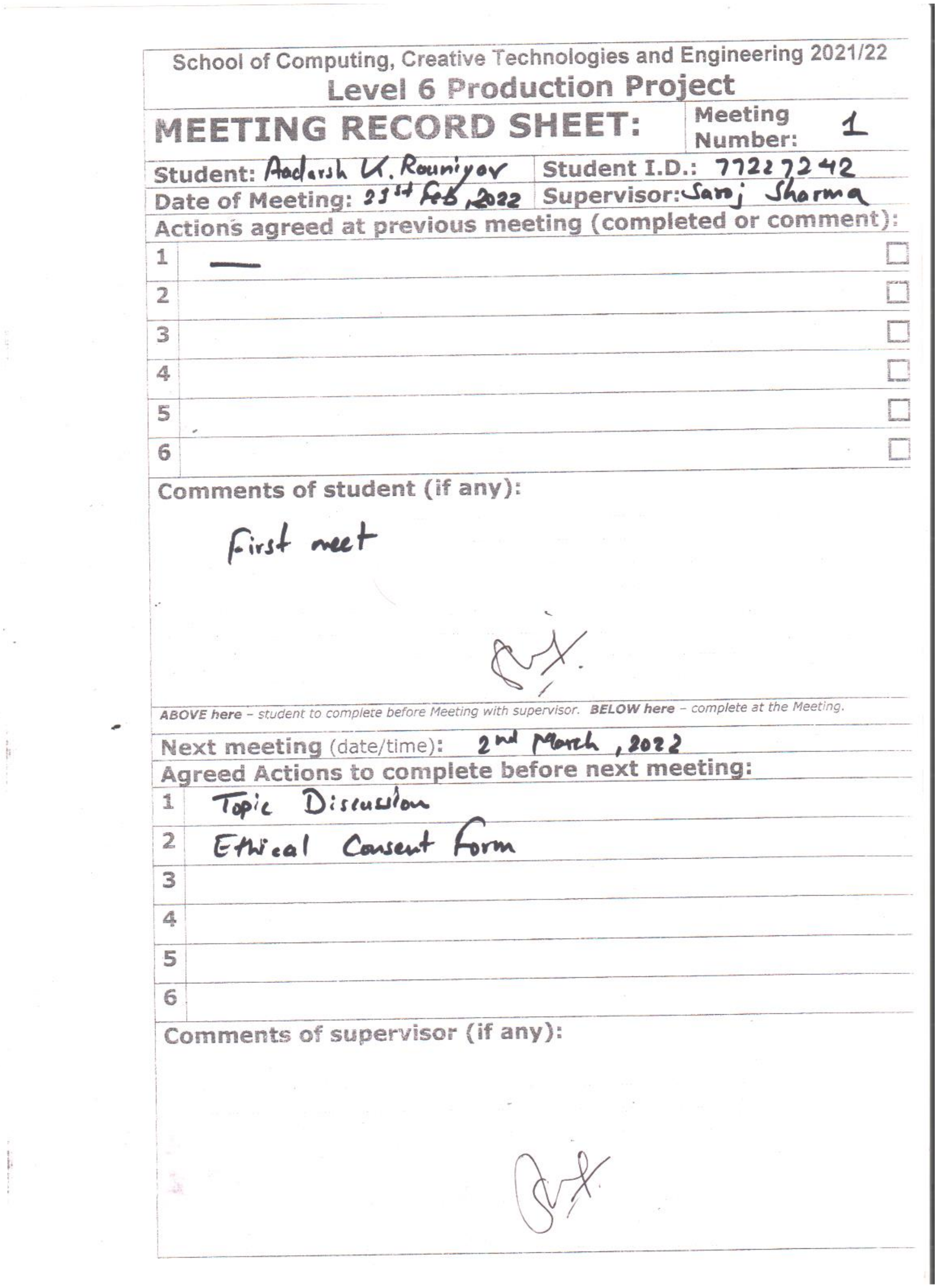
APPROVAL PROCESS

- Local Research Ethics Co-ordinator = LREC
- School level group (if your School uses a different review process, please follow your School guidance)
- University Research Ethics Sub-Committee = URESC

Category	Student applicants	Staff applicants
Risk Category 1	If your study has been provisionally classified as Risk Category 1, your Research Supervisor (or Director of Studies) can normally give approval for the project. You must complete this form and submit it to your Research Supervisor for consideration. A copy of the signed form if approved must be given or emailed to the LREC and, where relevant, the Research Module Leader, for information.	If your study has been classified as Risk Category 1, you do not need ethical approval for the project. You must complete the remainder of this form so that your research project is registered with the University. Please submit this form to your LREC.
Risk Category 2	If your study has been provisionally classified as Risk Category 2, your Supervisor (or Director of Studies) can recommend approval for your study by the LREC. You must complete this application form and also the separate Stage 2 - Research Ethics Approval form. Once you have completed the forms please submit both forms and supporting documentation to your Research Supervisor for consideration. Your Supervisor may disagree with your assessment and ask you to make revisions or reject your application. When the Research Supervisor is happy to recommend the application for approval, they will send the forms to the LREC. The LREC will review your project and then decide to approve it, ask for revisions, reject it or pass it on for review by the School level group.	If your study has been provisionally classified as Risk Category 2, your project will be considered for ethical approval by the LREC. You must complete this application form and also the separate Stage 2 - Research Ethics Approval form. Please submit both forms and supporting documentation to your LREC for consideration. The LREC will review your project and then decide to approve it, ask for revisions or pass it on for review by the School level group.
Risk Category 3	**Postgraduate Research Students** If your study has been provisionally classified as Risk Category 3, your Supervisor or Director of Studies can recommend approval for your study by the LREC. You must complete this application form and also the separate Stage 2 - Research Ethics Approval form and submit both forms to your Director of Studies. If your Director of Studies recommends approval of your project they will refer it to the LREC who will review your project and decide whether to grant ethical approval, request revisions, reject the application or refer it to the School level group for review. **Undergraduate and Taught Postgraduate Students** If your study has been provisionally classified as Risk Category 3, you should consult with your Research Supervisor immediately as it is unlikely you will be able to proceed and you should negotiate a project that is of lower risk. However, if you have already discussed the project with your Supervisor and they have agreed that a case for approval is warranted, proceed in line with the details above for Research Students.	If your study has been provisionally classified as Risk Category 3, your project will be considered for ethical approval by an appropriate LREC. You must complete this application form and also the separate Stage 2 - Research Ethics Approval form and submit both forms with supporting documentation to your LREC. The LREC will review your project and then decide to approve it, ask for revisions or pass it on for review by the School level group.
Q23	If question 23 has been answered 'yes', your application will be reviewed by the Chair of the University Research Ethics Sub-committee. The answer does not affect the Risk Category.	

Meeting Records

Meeting 1



School of Computing, Creative Technologies and Engineering 2021/22

Level 6 Production Project

MEETING RECORD SHEET: Meeting Number: 1

Student: Aadarsh K. Rouniyar | Student I.D.: 77227242

Date of Meeting: 23st Feb, 2022 | Supervisor: Saroj Sharma

Actions agreed at previous meeting (completed or comment):

1	—	□
2		□
3		□
4		□
5		□
6		□

Comments of student (if any):

First meet

ABOVE here – student to complete before Meeting with supervisor. BELOW here – complete at the Meeting.

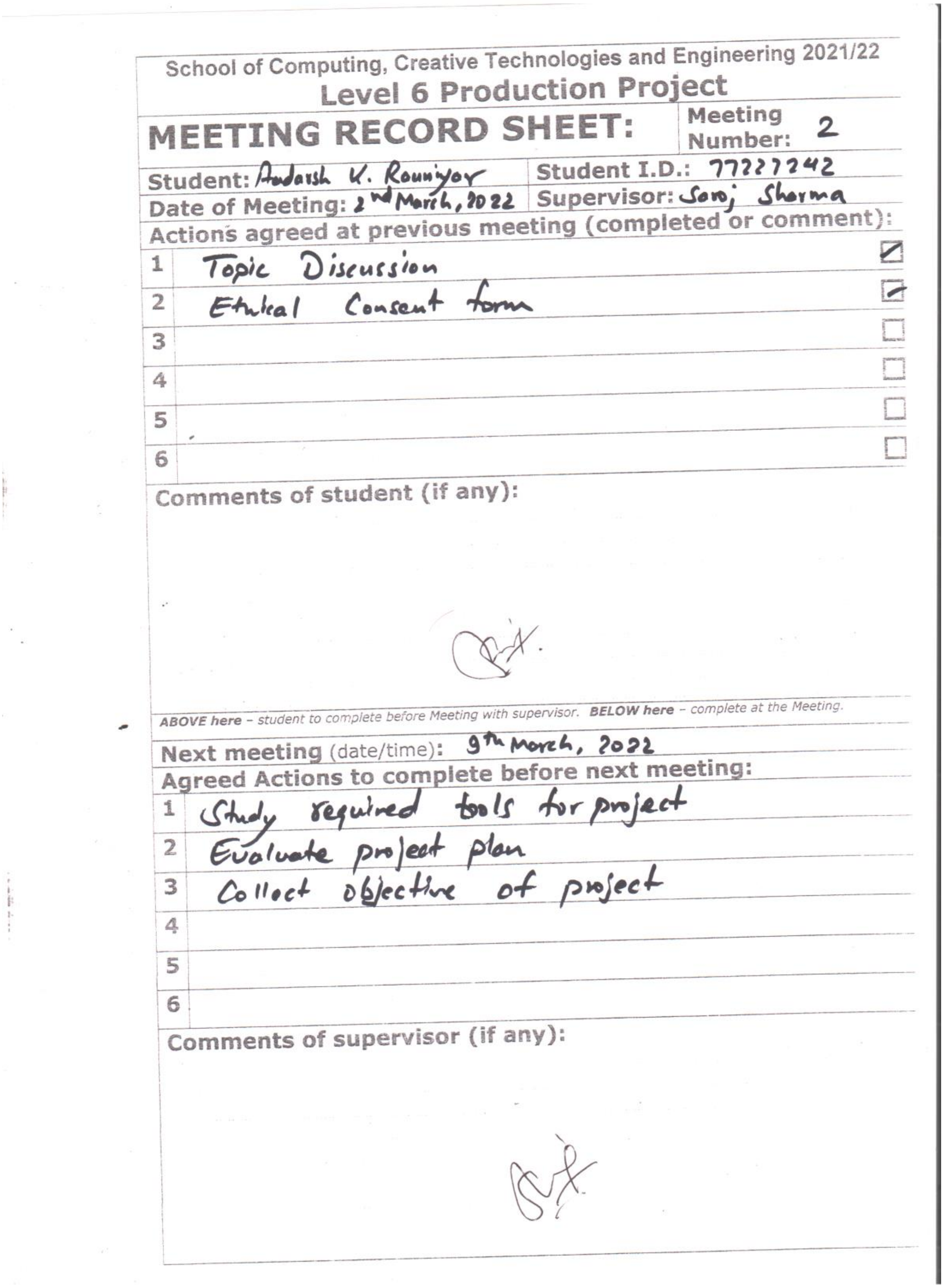
Next meeting (date/time): 2nd March, 2022

Agreed Actions to complete before next meeting:

1	Topic Discussion
2	Ethical Consent Form
3	
4	
5	
6	

Comments of supervisor (if any):

Meeting 2



School of Computing, Creative Technologies and Engineering 2021/22

Level 6 Production Project

MEETING RECORD SHEET: Meeting Number: 2

Student: Aadarsh K. Rouniyar **Student I.D.:** 77227242

Date of Meeting: 2nd March, 2022 **Supervisor:** Saroj Sharma

Actions agreed at previous meeting (completed or comment):

1 Topic Discussion ☑

2 Ethical Consent form ☑

3 □

4 □

5 □

6 □

Comments of student (if any):

***ABOVE here** – student to complete before Meeting with supervisor.* ***BELOW here** – complete at the Meeting.*

Next meeting (date/time): 9th March, 2022

Agreed Actions to complete before next meeting:

1 Study required tools for project

2 Evaluate project plan

3 Collect objective of project

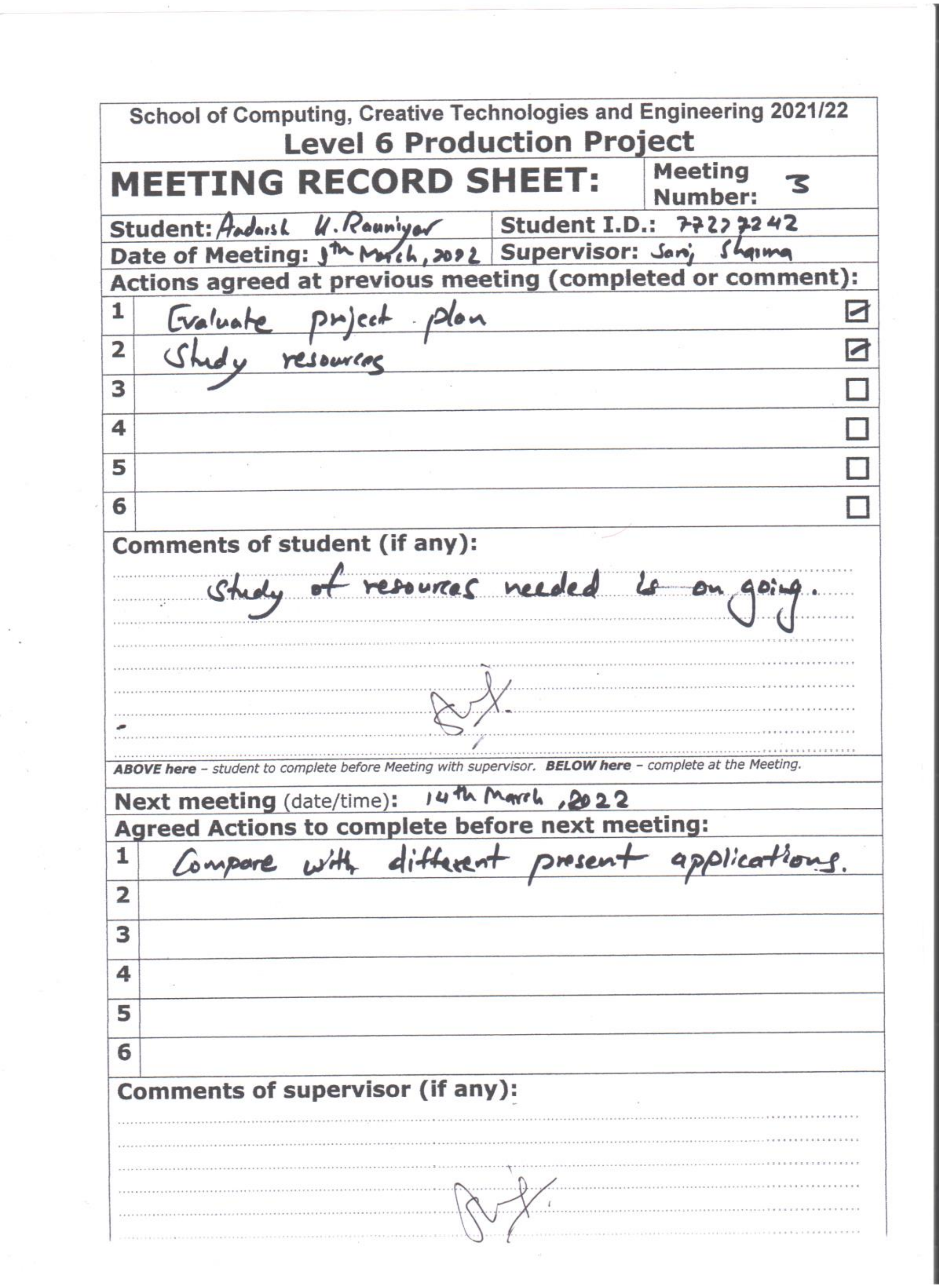
4

5

6

Comments of supervisor (if any):

Meeting 3



School of Computing, Creative Technologies and Engineering 2021/22

Level 6 Production Project

MEETING RECORD SHEET: **Meeting Number:** 3

Student: Aadarsh K. Rauniyar **Student I.D.:** 77227242

Date of Meeting: 9th March, 2022 **Supervisor:** Saroj Sharma

Actions agreed at previous meeting (completed or comment):

1	Evaluate project plan	☑
2	Study resources	☑
3		□
4		□
5		□
6		□

Comments of student (if any):

Study of resources needed is on going.

ABOVE here – student to complete before Meeting with supervisor. ***BELOW here*** *– complete at the Meeting.*

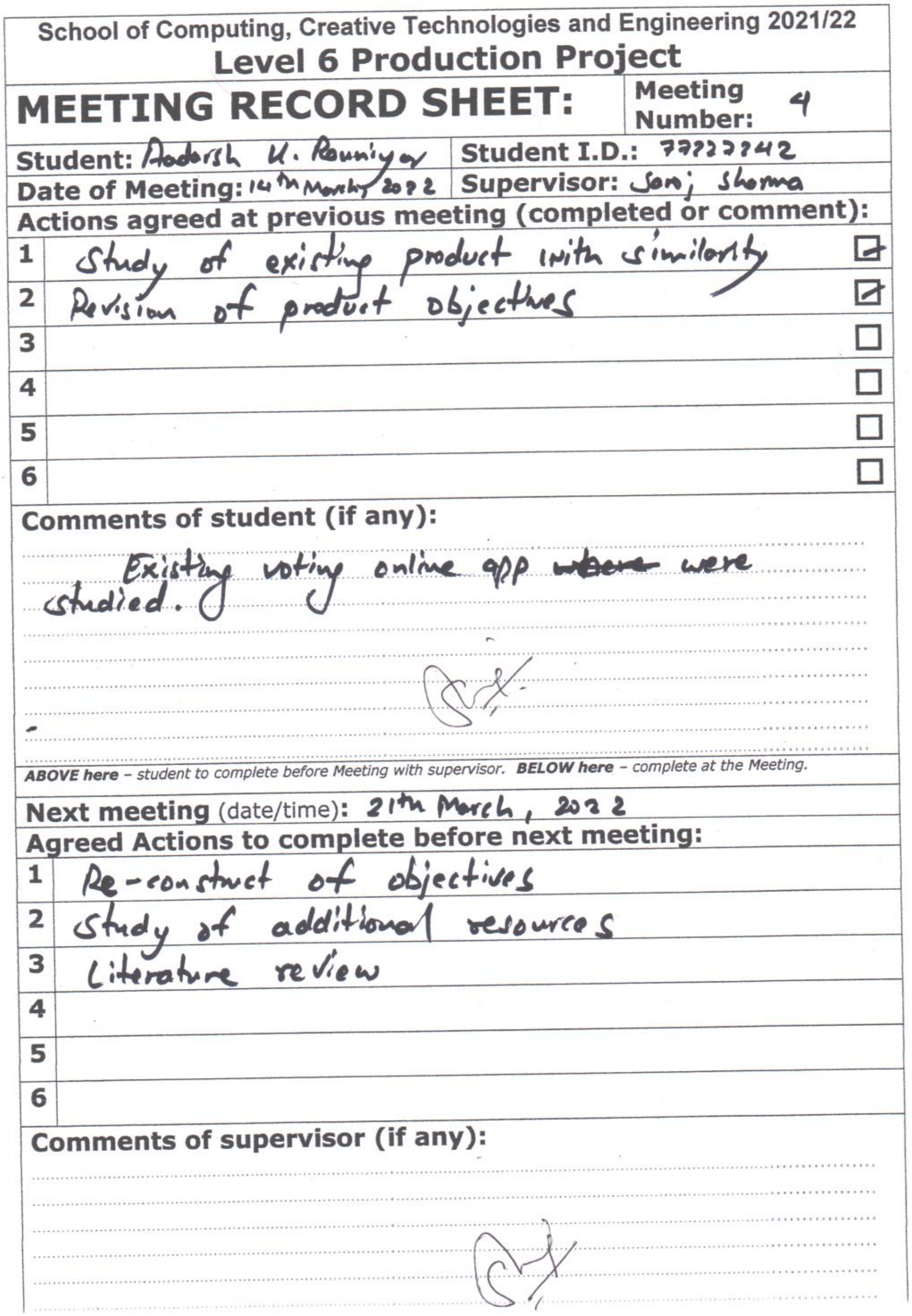
Next meeting (date/time): 14th March, 2022

Agreed Actions to complete before next meeting:

1	Compare with different present applications.
2	
3	
4	
5	
6	

Comments of supervisor (if any):

Meeting 4



School of Computing, Creative Technologies and Engineering 2021/22

Level 6 Production Project

MEETING RECORD SHEET: Meeting Number: 4

Student: Aadarsh K. Rauniyar | Student I.D.: 77227242

Date of Meeting: 14th March 2022 | Supervisor: Saroj Sharma

Actions agreed at previous meeting (completed or comment):

1	Study of existing product with similarity	☑
2	Revision of product objectives	☑
3		□
4		□
5		□
6		□

Comments of student (if any):

Existing voting online app ~~where~~ were studied.

ABOVE here – student to complete before Meeting with supervisor. BELOW here – complete at the Meeting.

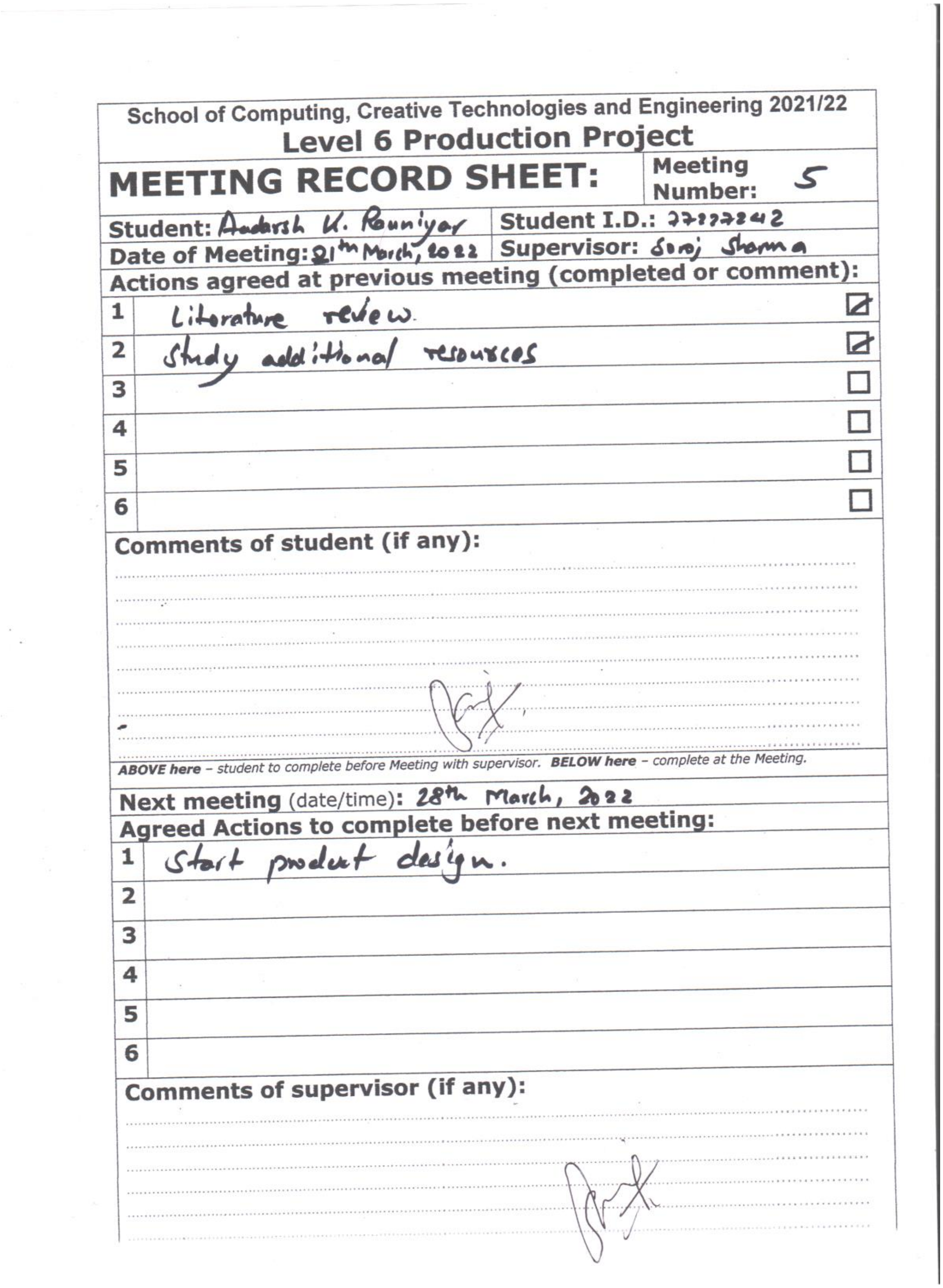
Next meeting (date/time): 21th March, 2022

Agreed Actions to complete before next meeting:

1	Re-construct of objectives
2	Study of additional resources
3	Literature review
4	
5	
6	

Comments of supervisor (if any):

Meeting 5



School of Computing, Creative Technologies and Engineering 2021/22

Level 6 Production Project

MEETING RECORD SHEET:	Meeting Number: 5
Student: Aadarsh K. Rauniyar	Student I.D.: 77227842
Date of Meeting: 21th March, 2022	Supervisor: Saroj Sharma

Actions agreed at previous meeting (completed or comment):

1	Literature review.	☑
2	Study additional resources	☑
3		□
4		□
5		□
6		□

Comments of student (if any):

ABOVE here – student to complete before Meeting with supervisor. BELOW here – complete at the Meeting.

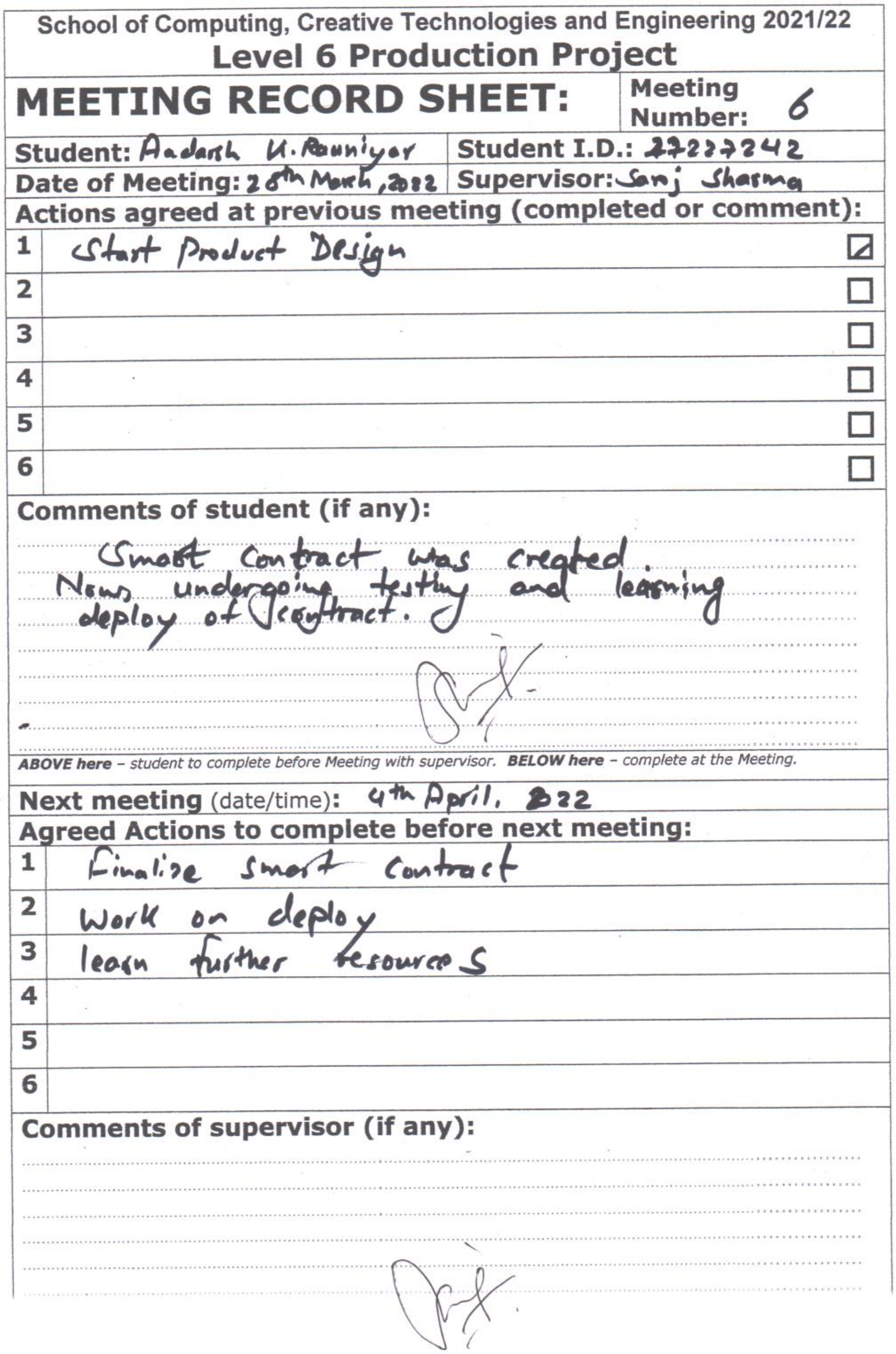
Next meeting (date/time): 28th March, 2022

Agreed Actions to complete before next meeting:

1	Start product design.
2	
3	
4	
5	
6	

Comments of supervisor (if any):

Meeting 6



School of Computing, Creative Technologies and Engineering 2021/22

Level 6 Production Project

MEETING RECORD SHEET:	**Meeting Number:** 6
Student: Aadarsh U. Rauniyar	**Student I.D.:** 77227242
Date of Meeting: 28th March, 2022	**Supervisor:** Sanj Sharma

Actions agreed at previous meeting (completed or comment):

1	Start Product Design	☑
2		□
3		□
4		□
5		□
6		□

Comments of student (if any):

Smart contract was created.
Now undergoing testing and learning
deploy of contract.

ABOVE here – student to complete before Meeting with supervisor. ***BELOW here*** *– complete at the Meeting.*

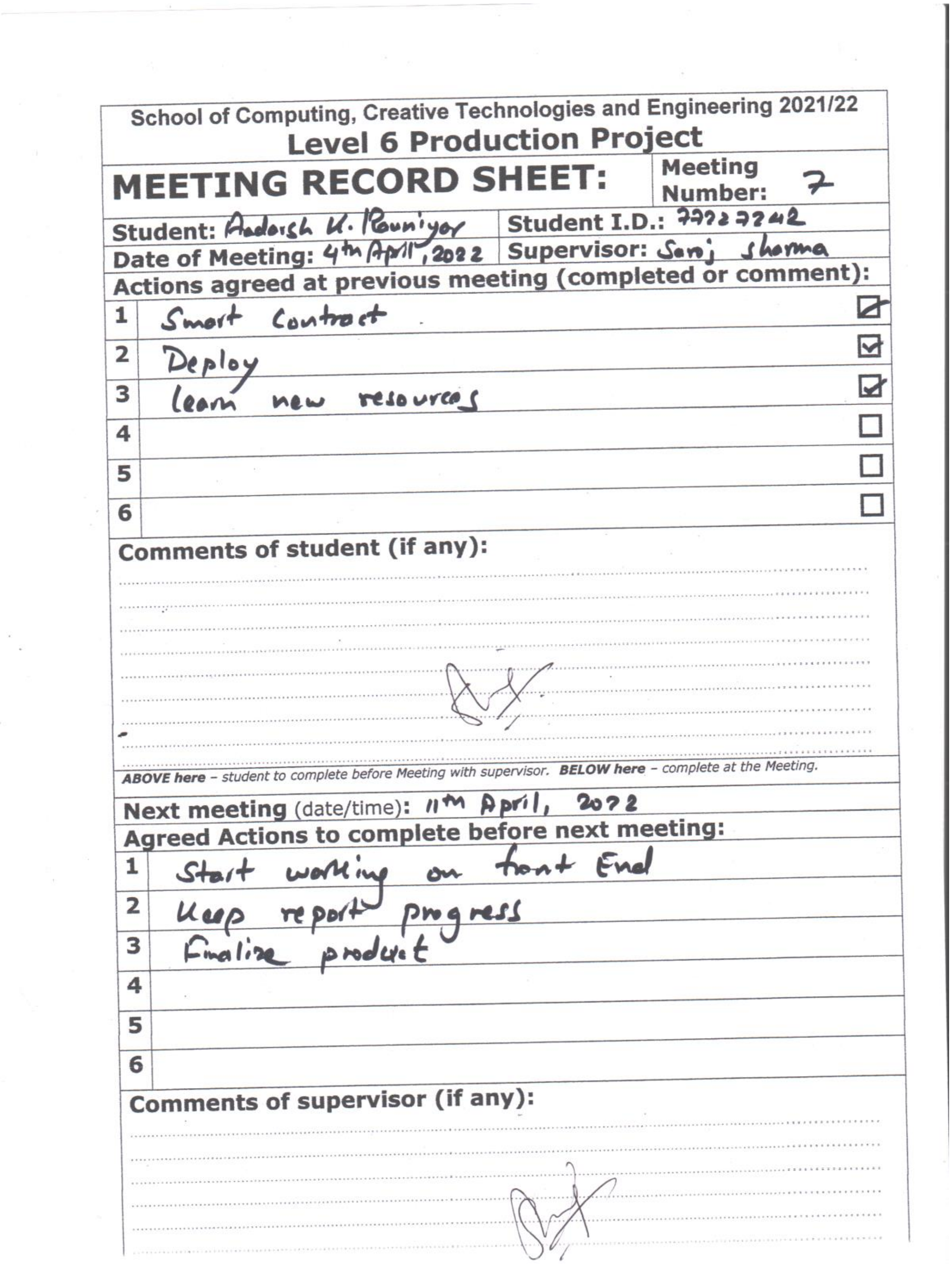
Next meeting (date/time)**:** 4th April, 2022

Agreed Actions to complete before next meeting:

1	Finalize smart contract
2	Work on deploy
3	learn further resources
4	
5	
6	

Comments of supervisor (if any):

Meeting 7



School of Computing, Creative Technologies and Engineering 2021/22

Level 6 Production Project

MEETING RECORD SHEET: Meeting Number: 7

Student: Aadarsh K. Rauniyar

Student I.D.: 77227242

Date of Meeting: 4th April, 2022

Supervisor: Saroj Sharma

Actions agreed at previous meeting (completed or comment):

1	Smart Contract	☑
2	Deploy	☑
3	learn new resources	☑
4		□
5		□
6		□

Comments of student (if any):

ABOVE here – student to complete before Meeting with supervisor. BELOW here – complete at the Meeting.

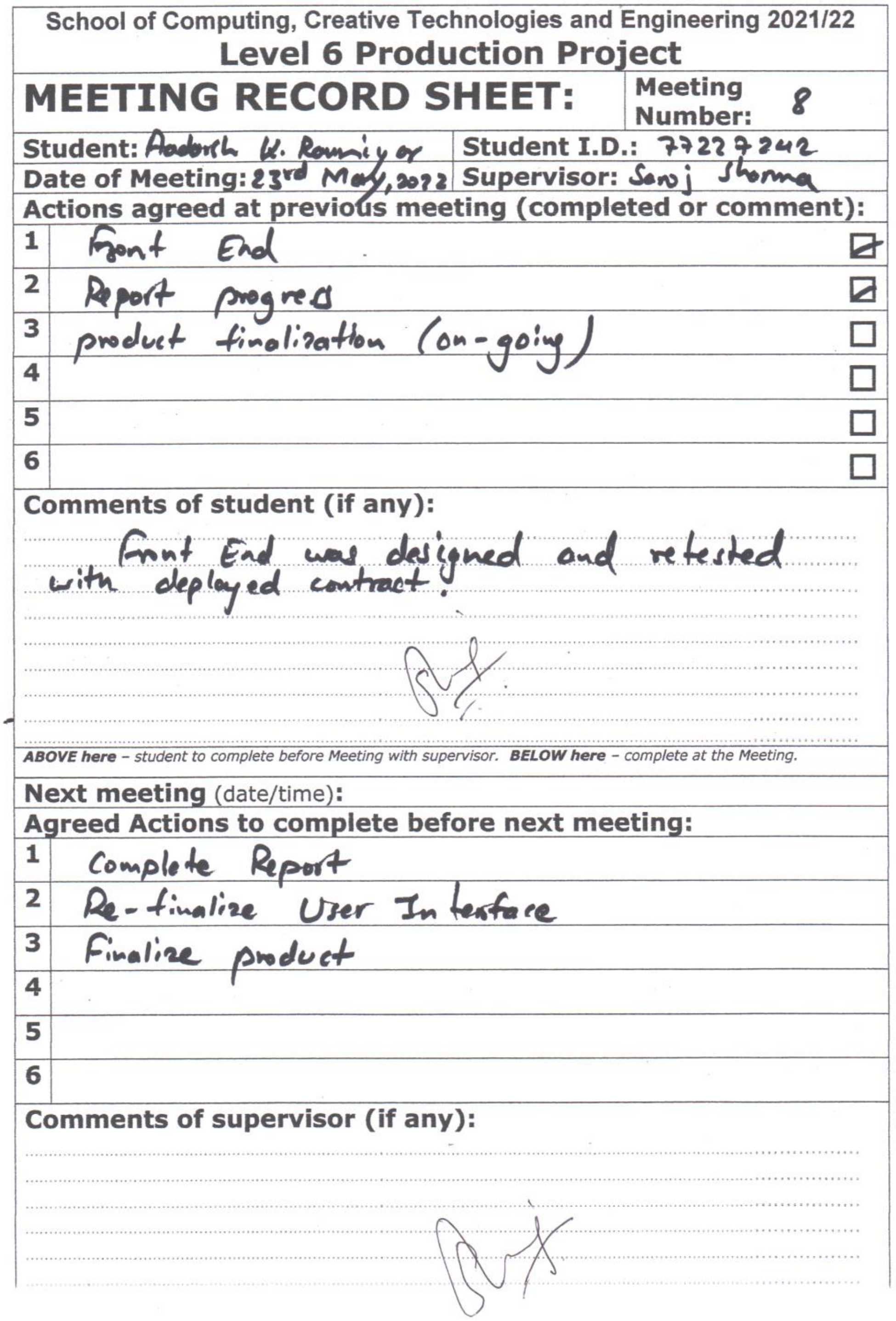
Next meeting (date/time): 11th April, 2022

Agreed Actions to complete before next meeting:

1	Start working on front End
2	Keep report progress
3	Finalize product
4	
5	
6	

Comments of supervisor (if any):

Meeting 8



School of Computing, Creative Technologies and Engineering 2021/22

Level 6 Production Project

MEETING RECORD SHEET:	**Meeting Number:** 8
Student: Aadarsh K. Rauniyar	**Student I.D.:** 77227242
Date of Meeting: 23rd May, 2022	**Supervisor:** Saroj Sharma

Actions agreed at previous meeting (completed or comment):

1	Front End	☑
2	Report progress	☑
3	product finalization (on-going)	□
4		□
5		□
6		□

Comments of student (if any):

Front End was designed and retested with deployed contract!

***ABOVE here** – student to complete before Meeting with supervisor.* ***BELOW here** – complete at the Meeting.*

Next meeting (date/time)**:**

Agreed Actions to complete before next meeting:

1	Complete Report
2	Re-finalize User Interface
3	Finalize product
4	
5	
6	

Comments of supervisor (if any):