

Graduation Project

At the “notJustDevelopment” company

Thesis document



Creator: Vladyslav Petrykov

Student Number: 471157

Email: 461157@student.saxion.nl

Phone: +380972307557

Document Version: 1.5



Table of Contents

<i>Version history</i>	4
<i>Acknowledgments</i>	6
<i>Abstract</i>	7
<i>Terminology</i>	8
<i>1. Introduction</i>	9
<i>2. Problem statement</i>	10
<i>3. Case analysis (stakeholders, requirements)</i>	12
3.1. Stakeholders	12
3.2. Requirements.....	14
3.3. Possible solution	16
<i>4. Governance (project/process management)</i>	17
4.1. Project management	17
4.2. Meetings	17
4.3. Workflow management	18
4.4. DevOps Platform.....	19
4.5. Quality assurance.....	19
<i>5. Research</i>	20
<i>6. Design</i>	22
6.1. Architecture design	22
6.2. Lo-Fi design stage	24
6.3. Hi-Fi design stage	25
<i>7. Proof of concept</i>	27
<i>8. Realization</i>	30
8.1. AWS Amplify	31
8.2. Solidity	33
8.3. React.....	34
8.4. Ganache	35
8.5. Metamask.....	36
8.6. Ropsten.....	37
8.7. Frontend testing	39

8.8. Frontend deployment.....	40
8.9. Backend testing	41
8.10. Backend deployment.....	41
8.11. Setup possibilities	42
8.12. Pipelines	43
9. <i>Validation</i>	44
9.1. Tests.....	44
9.2. Robust	44
9.3. High quality	45
10. <i>Conclusion</i>	46
11. <i>Bibliography</i>	47

Version history

Version number	Delivery date	Content
Vo.1	15.03.2021	<ul style="list-style-type: none">- Introduction- Problem statement
Vo.2	25.03.2021	<ul style="list-style-type: none">- Case analysis- Requirements- Governance
Vo.3	01.04.2021	<ul style="list-style-type: none">- Research- Design- Architecture- Lo-Fi
Vo.4	03.04.2021	<ul style="list-style-type: none">-Hi-Fi
Vo.5	10.04.2021	<ul style="list-style-type: none">-Proof of concept
Vo.6	15.04.2021	<ul style="list-style-type: none">- AWS Amplify- Solidity- React
Vo.7	22.04.2021	<ul style="list-style-type: none">- Ganache- Metamask
Vo.8	29.04.2021	<ul style="list-style-type: none">- Ropsten- Frontend testing- Backend testing
Vo.9	06.05.2021	<ul style="list-style-type: none">- Frontend deployment- Backend deployment `
V1.0	13.05.2021	<ul style="list-style-type: none">- Code & Product quality
V1.1	20.05.2021	<ul style="list-style-type: none">- Setup possibilities- Pipelines

V1.2	23.05.2021	- Validation - Tests
V1.3	23.05.2021	- Robust - High quality
V1.4	26.05.2021	- Conclusion - Appendices
V1.5	27.05.2021	- Bibliography

Acknowledgments

Since the beginning of this project, the war in my country has begun. I want to thank Saxion teachers who supervised this project and company members for their support during this challenging time. I appreciate any physical and mental help from everyone related to this thesis.

Thanks to D.Heijnk for being able to meet weekly and for such good conversations that significantly impacted the project. I would also like to thank my company supervisor Vadim Savin for all the time involved in the project.

Abstract

To start with, this is a project that started from an idea only. From the processes management and code development point of view, I have tried to involve all relevant knowledge I have studied at Saxion. Next, I have researched and added some techniques/technologies found on my own. I combine all my skills in a single project, resulting in this document.

I want to clarify the design stage where I had to think of UI / UX patterns for this project. In general, my study indirectly concerns design at all. I have tried to combine reached knowledge to develop good design choices.

Another essential thing to mention is that I carried out the project myself. This relates to the chosen professional activities (Analyze, Design, Realization).

The project is exciting because it handles data in a mix of using already well-known centralized ways of handling data and decentralized at the same time. I have created different diagrams and included screenshots to give a reader of this document a clear understanding of all processes. I have tried not to overload a reader with useless data in this document, so some chapters reference the attached documents.

Terminology

NFT	NFT is a unique token representing art, music, videos, video-games objects, etc. Each bought NFT serves as proof of ownership. It has been available since 2014 but has become increasingly popular because it is used to buy and sell digital artworks.
Gas fees	It is a compensation paid to the so-called “miners.” Their hardware participates in verifying transactions and securing blockchain network.
Minting	It is a process of creating or publishing something on the blockchain.
Blockchain transactions	A decentralized way of communication with blockchain. A single transaction could contain a price, asset, or ownership. Once the transaction is complete it is recorded and stored in a decentralized manner.
MetaMask	An extension to authorize users to the EVM-based websites using a crypto wallet.
Amplify Studio content	A webpage in Amplify Studio where all records of a particular model are stored.
dApp	It is a decentralized application that is running on a decentralized network. It combines the usage of smart contracts and a frontend user interface.
Smart contract	It is a computer program that automatically executes all parts of a contract and stores data on a decentralized network.
Web 2	Refers to the version of the internet It is dominated by companies that provide services and controls your personal data. All data flow goes through centralized servers.
Web 3	Refers to the decentralized applications that run on the. Blockchain. The data flow cannot be hacked, the data is highly secured.
EVM	Ethereum Virtual Machine – the heart of the Ethereum blockchain. It deploys different functionalities to the blockchain that developers can use.

1. Introduction

The graduation assignment will be realized at notJust Development, registered on Calle Agustin Espinosa Garcia 4, SC DE Tenerife, Spain. notJust Development is a young educational company that is helping people all around the world to learn full-stack development. A recent Saxion Graduate started the company, and it grew to 5 employees in the first year. The team works remotely, and three professional software engineers are in the IT department. The free educational lectures created by notJust Development reached 2.7 million people worldwide.

The graduation assignment will be realized at notJust Development, registered on Calle Agustin Espinosa Garcia 4, SC DE Tenerife, Spain. notJust Development is a young educational company that is helping people all around the world to learn full-stack development. A recent Saxion Graduate started the company, and it grew to 5 employees in the first year. The team works remotely, and three professional software engineers are in the IT department. The free educational lectures created by notJust Development reached 2.7 million people worldwide.

I will join the remote team at notJust Development in an IT department to get experience in web3 technologies by designing and building a dApp. I will be introduced and helped get started with all the new technologies involved in the project. At the same time, I am the only developer in the IT department who will deal with web3 research and implementation.

I am targeting this document to the teachers and commissioner who will grade my thesis. At the same time, I have adopted chapters for readers who want better to understand the decentralized software development way of work.

2. Problem statement

It is straightforward to fake a digital certificate. It may complicate the hiring process for other companies because they must prove that the applicant's certificate is valid and relevant. For an HR person, it is a long process to go through. Nowadays, companies blindly trust future employees on their completed courses.

notJust Development provides education with premium educational academies. When users finish the course, they want to attach a digital certificate to the course graduator. This process is accomplished by sending a .pdf certificate via email. This file includes information about the graduator with the relevant course that has been recently completed. It takes much time for a manager to create and send those files to the graduates.

Based on the mentioned processes/issues described above, the company wants to solve them with an automated solution. The project's desired outcome is to have a web application that will allow a course teacher to create and issue digital certificates. In this way, graduates will validate their ownership by using blockchain technologies. The main goal is to create a secure platform where any representative administrator of the company can issue certificates as NFT in a decentralized manner.

The project is mainly structured around a blockchain, dApp development, and NFTs. The technologies are modern and new to the society. Different variations can be observed in combining, developing, and releasing a product. Each of the existing blockchain programming languages, supporting tools, and deployment platforms has its downsides and benefits. Since one of the project's main goals is to build a functioning, cost-efficient product that matches the company's needs for validating future graduates, I will do proper research with reasonable conclusions.

The main research question is **“How to integrate a certification process via blockchain.”** In the list below, I will put other established research questions given by the company. Those questions are the priority to be answered during this stage. At the same time, they were divided into sub-questions within each research paper. They can be found in the attached research papers to this project. They include investigation, analysis, and comparison stages. It is essential to clarify that discovering unexpected topics may be observed during the research stage.

- How to develop and release an application with a core usage of Web3 blockchain methods?
- What is an ERC-721 Non-Fungible Token Standard, and how to create one?
- What is the best blockchain platform for our use case? (Ethereum, Solana, Polygon)?
- How to prove ownership of the NFT's that the client owns?
- How to design the Web3 application in a clear and user-friendly way?

The company expanded the project scope during the research phase. Additionally, notJustDevelopment wants to support small companies and open source projects. The range of that support is represented in the additional functionality of buying and selling items on our platform. Small companies could expand their budget and audience by selling non-fungible tokens to other users. Software must give the same functionality to all users on the platform. From the user's perspective, they can support their favorite open source projects by buying these NFTs and showing their support to the world by holding the NFT. Later, if users decide, they can sell the NFT to other users on the platform for a profit.

3. Case analysis (stakeholders, requirements)

3.1. Stakeholders

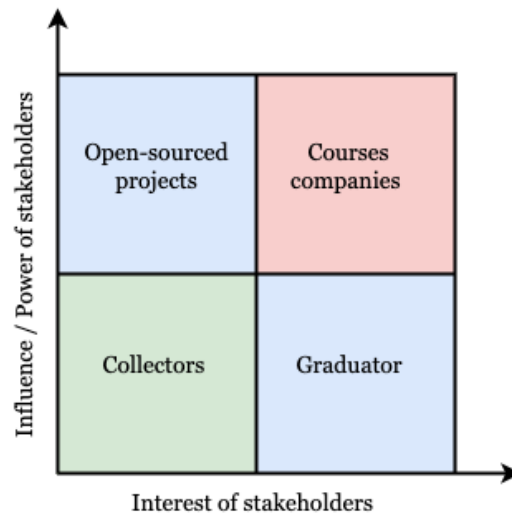
Multiple parties are interested in a product and possibly can affect or be affected by the released functionalities.

The primary stakeholder of this project is the company itself. As mentioned in the previous chapter, certification is the main problem after the course finishes. A person from the company must be able to replace current certification processes by using the released software I developed. This means there must be a way of creating certificates as NFTs according to the requirements. At the same time administrator must be able to assign the assembled product to the course graduator.

The second stakeholder of this project is the graduator itself. There must be a proper way of retrieving and displaying certificates for the user. The main goal for this stakeholder is to have a way of proving the ownership of the related NFTs. It may impact the recruiting process for future employees by giving a clear understanding of the completed courses. At the same time, the certificate owner must have an option to put their item on sale. The card will permanently save the graduator's details, and the future owner won't be able to use it. An NFT may grow in price and become a valuable selling item. For the NFTs representing a certificate, buying and selling don't bring value because the NFT has value only for the original owner.

However, for other types of NFTs unrelated to credentials, buying and selling bring much value to all the parties. It is a third stakeholder within a product - collectors. For example, user John loves an open-source library called "cute-kitties" and uses it. Cute-kitties decide to crowd-source the release of a new major version that requires much work from the developers. They sell 100 stickers for \$100 each, using the collected money to fund the latest version. In exchange for his support, John received a sticker with cute kitties and specifically with that version that he supported. After two years, the cute-kitties library grew in popularity, and all the big companies started using it. John's sticker for 100 dollars to support the open-source project is worth \$1000. He decides to sell it and makes a profit. Everyone is happy. The library was able to crowdsource a significant release that contributed to its success. The user was able to support the company and increase the list of their collectibles.

The fourth party involved in this project is open-source projects. They will be able to publish items on sale directly. Those could be some unique arts or assets that companies are using. If the created item were sold at the defined price, the company would gain all its value on its crypto wallet.



img. 0.1 – Stakeholders interest & influence

The image above (img 0.1) represents a stakeholder interest in this project. As it can be seen, the highest power and interest are focused on course companies. Their claim is very high in replacing the basic graduation system with a decentralized certification.

Then graduates show an increased interest in retrieving those certificates and being able to prove their knowledge using the developed platform.

The strong influence of open-sourced projects directly relies upon the selling items they will publish. Because of such actions, the platform activity will be increased. It will impact both NFT prices and the platform activity.

And lastly, collectors are the stakeholders who do not have a considerable interest in a project. They usually track multiple marketplaces simultaneously, looking into collectibles they are interested in. They do not attach a high interest to a specific platform but only to a product.

3.2. Requirements

Initially, the entire project held an idea of what processes must be integrated. The requirements were needed before starting the project. Requirements were structured using a MoSCoW technology that denotes a category of prioritization.

Firstly, the project gained an empathetic understanding of the client's problem. After this, I have defined requirements based on the meeting's feedback with the contact person. This stage gives a clear vision of an accurate problem statement from the client's perspective. Following the design thinking method, the next step is to ideate defined requirements and identify innovative solutions to the problem. The prototype stage is an essential step in changing existing and releasing additional requirements until the final version of the requirements is complete. Finally, the design thinking method has a testing step that helps understand if the requirements are reasonable enough or must be changed.

Establishing requirements at the beginning of the project gave a clear vision of what a specific party within the project must be able to do — at the same time, providing a clear understanding of the entire system. In the end, I started the project with the objective and smart requirements. There is a distinction between main system requirements and functional and technical. The system requirements can be achieved when the product is developed, matching functional and technical ones.

System requirements

- **R1:** A Crypto sticker can be used as a certificate (ex: AWS certificate, google certificate, certificate of completion the course X)
- **R2:** Crypto stickers can show support for a company/open-source project/content creator.
- **R3:** Crypto stickers can be used as access to a community (for example, everyone who owns the sticker X has access to a private community)

Functional requirements

- **F1:** The administrator must be able to upload/create a crypto sticker with the following information:
 - The sticker **image** (png), **name**, **description**, **issuer** (the organization that is issuing this sticker), **price**
- **F2:** The administrator must be able to issue a crypto sticker to a specific user.

- **F3:** A user must be able to receive crypto stickers
- **F4:** A user must be able to mint a crypto sticker
- **F5:** A user must be able to see a list of all his crypto stickers
- **F6:** A user should be able to see the details about one sticker (all metadata and the history of its transaction)
- **F7:** A user must be able to sell a crypto sticker
- **F8:** A user must be able to see all crypto stickers that are for sale
- **F9:** A user must be able to buy a crypto sticker from the marketplace
- **F10:** A user should be able to share his sticker collection, and anyone can view his collection
- **F11:** A user could sell only sellable stickers (for example, he cannot sell a certificate sticker)

Nonfunctional requirements

- **NF1:** The system must be deployed to a reliable and cost-efficient blockchain
- **NF2:** The code should be tested, and the code coverage should be more than 90%
- **NF3:** Any read operations must not take longer than 3 seconds
- **NF4:** Any write operations (to the ledger) must not take longer than 1 minute
- **NF5:** To improve security, only the administrator must be able to create stickers
- **NF6:** After the sticker is issued, it should be immutable:
 - The most critical data about a sticker should be written to the blockchain
 - The png images for the stickers must be stored in a decentralized datastore
- **NF7:** When a sticker is minted, the price will be split into:
 - 96.5% goes to the issuer
 - The platform will keep 3.5% transaction fees
- **NF8:** When a sticker is sold on the marketplace, the price will be split into:
 - 96.5% goes to the seller
 - The platform will keep 3.5% transaction fees

3.3. Possible solution

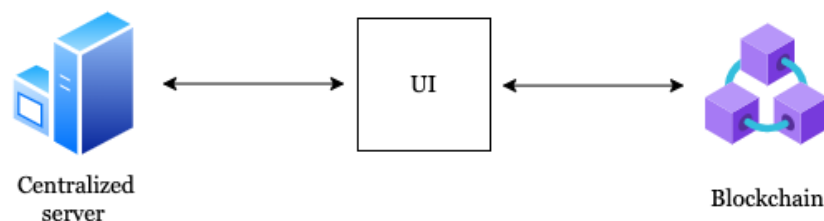
In this chapter, I will discuss technical solutions available to achieve product needs. Several approaches are available by following Web2 and Web3 technologies and combining those.

First, it is possible to store all data in a decentralized manner on a blockchain. On the one hand, it is a reliable way of storing information. The request processing speed is very high, with almost zero risks of losing it over a network. On the other hand, not all blockchain actions are free, which is not a cost-efficient way of handling requests from the client size.

Secondly, I can develop the platform by combining Web2 and Web3 technologies. MUST requirements will use the web3 technology stack. All operations are going to be achieved via blockchain across different nodes. For example, a flow of selling items will use both technologies. For listing items on the marketplace, centralized servers will store data, but buying operations will be processed via blockchain. Some other functionalities may also follow a similar flow as a listing explained in this chapter.

Lastly, the development can be done fully centralized. From the technical point of view, it is possible, and many well-known stacks are available. But this approach does not support NFTs usage. Because of this, further research activities will not be structured only around these technologies.

Indeed I will use external libraries to develop components, functionalities, etc. Whenever external library usage occurs, it will be adopted to match the design and functional choices of the system. The most plausible solution is shown in the image below (img 0.2). Based on the reasonings in this chapter, the application will follow this flow on handling data. It is important to mention that it will not be recognizable how the platform handles data from the user's point of view.



img. 0.2 – Possible communication architecture

4. Governance (project/process management)

4.1. Project management

The way of working for this project is Agile. There are several reasons for that. First of all, it has sprints that make it easier to divide a project into smaller parts. At the same time, it puts a clear vision on task estimations and time management. And finally, the company supervisor will see project results regularly by the end of each sprint. We established the duration of every sprint as two weeks. By the end of each sprint, we will have a sprint review. I will clarify all findings, solutions, and implementation choices for a particular task by presenting the sprint results.

The next meeting after the sprint review would be a retrospective. We will go through the past sprint to find possible improvement points. Next, we will start a new sprint planning session to create/prioritize tasks for the next sprint.

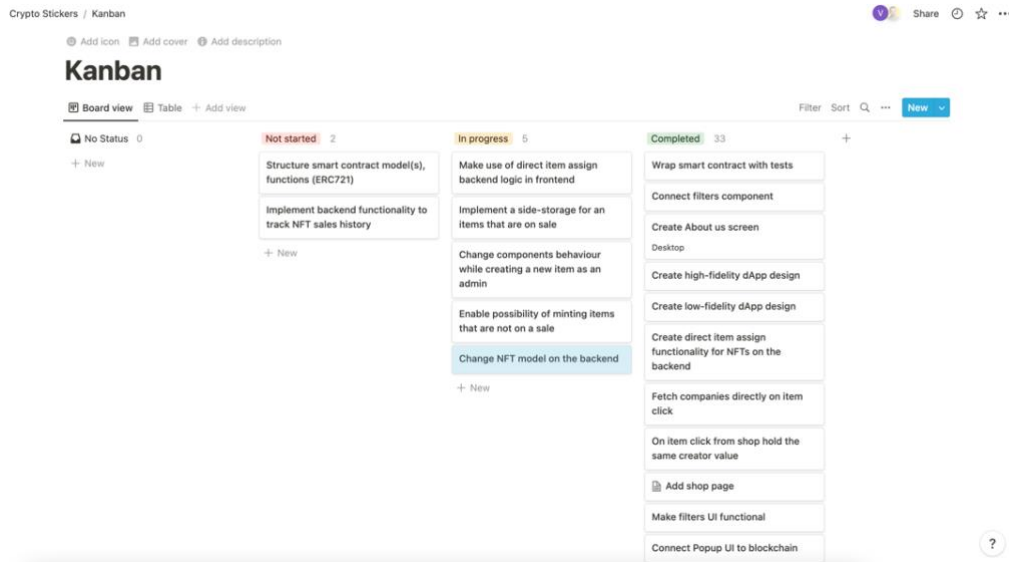
4.2. Meetings

I have established a text communication with a company supervisor via WhatsApp. This way, we will handle quick questions that we can solve without voice calls. At the beginning of the project, we agreed on regular calls every Monday and Thursday to quickly go through the active tasks. This meeting will take place on the Microsoft Teams platform. We decided to inform each other if we could not attend the scheduled call. In this case, the meeting will be rescheduled for another time or day.

The meetings with a teacher will take place every Friday via Microsoft Teams. Weekly calls primarily focus on showing my progress at the current diploma stage. I will update a teacher on all research, development, deployment, and documentation progress. All meetings are scheduled, but sometimes, we could skip and reschedule a call for another day.

4.3. Workflow management

One of the useful web applications that will simplify the project management for both the company supervisor and me is a Notion. This tool manages Kanban boards where a single project can be divided into several columns. An example of our Kanban can be seen in the image below (img 0.3).



img. 0.3 – Kanban board

Those columns are stages of a particular task. Starting from the backlog, followed by the not started, in progress, and completed columns. We brainstorm the functionalities with a high priority to do during the sprint planning session. After that, we measure and establish the number of tasks that are possible to complete for me within one week. Those cards are passed forward to the “Not Started” column. After that, I am responsible for ensuring that all the cards are completed within the estimated sprint time. Before passing a card under the “Completed” column, the code review session takes place with a developer from the company. Once the code quality is released, a new feature is merged into the master branch.

Overall, Notion is an application that provides components such as databases, Kanban boards, wikis, calendars, and reminders. Users can plug in these components to build knowledge management, note-taking, data management, project management, etc. (Wikipedia, 2022)

4.4. DevOps Platform

I will use a git-based open-source distributed version control system to manage the project's code. The DevOps platform that I will use for these purposes is GitLab. After the initialization, I will disable direct pushes on the master branch. At the same time, the way of working with GitLab will be feature-branching. The main principle is to copy the existing code on the master and work only on a single feature until complete. After the development stage, the code goes through refactoring, and then the developer will push it on the branch related to the implemented attribute. Next, I will establish a peer review session to explain and hear my implementation's possible improvements or downsides. If the peer review went well, the branch is ready to be merged into the master, where all the previously implemented features exist. I will carry out this process to ensure incoming functionalities do not break the existing product.

I will write several pipelines besides using GitLab as a web-based Git repository. The detailed information about pipeline scripts can be found in Chapter 8.12. Pipelines.

4.5. Quality assurance

One of the essential aspects of the project is readability and maintenance. Since I am the only developer on the project, I have applied air-BnB eslint code standards and prettier extensions. Using air-BnB code standards is based on the company's choices. It increases both readability and maintenance at the same time.

For example, when developers finish a task on a separate branch, they must execute several commands to prove that code applies to the defined eslint code standards. After that, the following code review process can go. So, following these techniques, I can better organize the product maintenance. If the system requires a change from the developer, those would be easy to integrate due to the similar applied code styles and structure across the entire code. Talking about prettier, it only enforces a consistent code style across a whole project's codebase, impacting the readability aspect.

5. Research

At the beginning of the project, I will analyze and dive into a research question on a raised topic. I will work on research questions by reading blogs, forums, and available information about involved technologies in this project. All research findings and conclusions will be separated and structured in different files. In the end, the research findings results will directly resolve created requirements.

It is essential to mention that the research stage will encounter many random questions due to the complexity and novelty of technologies. After answering a single question, it may result in additional unexpected research.

Brief answers to the main research questions can be found below:

1. NFT (minting, digital wallet, marketplace, uploading files, setting up sale processes):

The NFTs themselves will be images that represent a specific completed course for a graduator. At the same time, NFTs could represent the unique art as images created by open-source libraries. They will be minted on the blockchain and attached to the specific crypto wallet id. The sale process is followed by setting a special price on a particular NFT. And finally, to accomplish any action, the user must use the MetaMask digital wallet to achieve their needs.

2. dApp (communication, available languages, data storage):

I will set up the communication within an entire system using blockchain and web3.js libraries that enable smart contract interaction and data exchange. Talking about programming languages, I will use Solidity to develop backend functionalities and React to build a user interface. The non-centralized data storage will be a blockchain, and I will use Amazon servers for storing data in a centralized manner.

3. platform (smart contract - what is it, how it works, backend development/production modes, how and where to deploy)

The smart contract has defined conditions executed whenever it is called. The development and production modes are almost identical and differ in running blockchain mode. The web3.js either makes use of a local environment or a deployed one. The deployed version of the smart contract exists on the distributed nodes that communicate with each other and formulates a chain of data.

Before starting the actual research, I divided requirements into directly related topics. Because technologies in this project are new to me, I wanted to avoid an analysis-paralysis. It could have led the company to unexpected postpones in the planned scope of the project.

During this stage, the scope of the project expanded with additional requirements. Those are related to the buying and selling processes on the blockchain. In this way, the product comes from a different point of view, and we discussed it with the product owner. As it turned out, they wanted to extend an application in the future so it could act as a marketplace where startup companies can buy and sell items to expand their budget. At the same time, there would be a distinction between certificates and NFTs on sale by their name, description, and item type.

Introducing selling and buying functionalities comes only with benefits for all stakeholders. Administrators can digitally certify course gradulators on the completed course. Clients can further sell those certificates because they may grow in price after some period. The buyers are always interested in the process of collecting stickers, so they will be willing to get those certificates or regular NFTs items from companies. In case the owner of the certificate does not correspond to the wallet id, it won't be possible to use this NFT to prove course graduation. Each certificate contains a name and a corresponding company description that assigns those certificates. In this way, buying and selling functionalities only come from the collecting perspective and are unrelated to the knowledge base received in the courses.

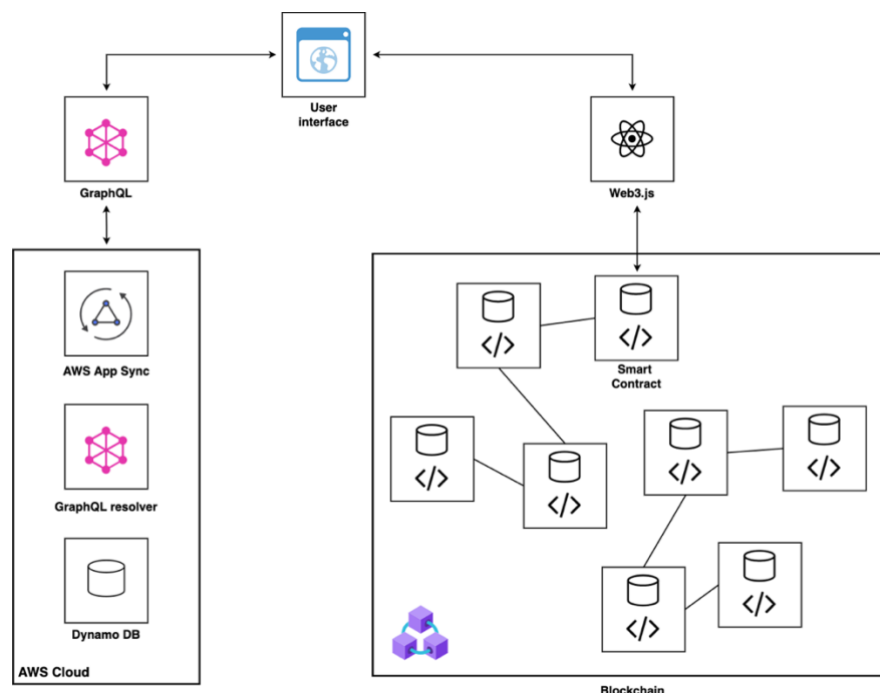
6. Design

Design question comes into several disciplines for this project. First, I must design a project flow and connection between different technologies. I had started this stage by the time when the proper research with conclusions was ready.

6.1. Architecture design

An entire system will be based on Web2 and Web3 technologies, according to my analysis in Chapter 3.3. Possible solution. Web3 collects all data on so-called blocks to hold sets of information. When the block is opened, it fills in the data inside it. Once the space of a single block is used, the blockchain closes this block and links to the previously filled block. In this way, it forms a chain of accessible data. Everything works in a decentralized manner. At the same time, I will establish centralized storage communication to store and retrieve data using API calls.

As shown in the image below (img. 0.4), I have established the connection from the frontend side via the web3.js library. Further usage of this library will allow me to connect to the smart contract. This contract will be made and deployed by me. It will include predetermined conditions to execute when called.



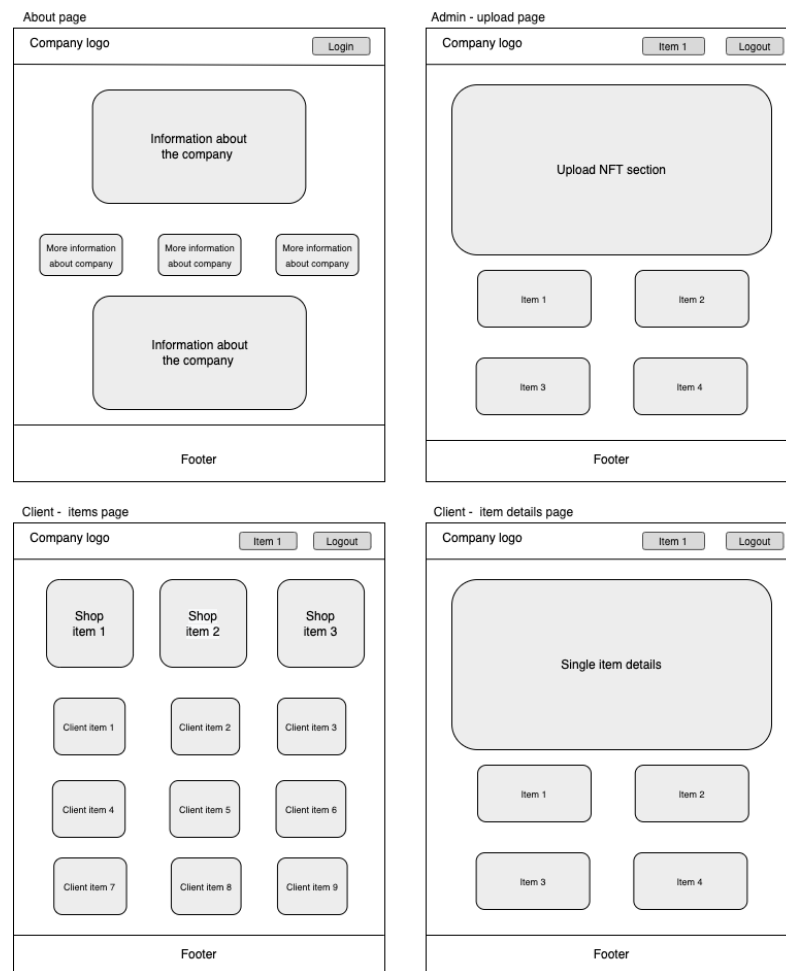
img. 0.4 – Architecture design

At the same time, on the left side of the diagram above, usage of the AWS services can be spotted. From the technical point of view, there are several technologies involved. In my case, I am using GraphQL to make fast, flexible, and developer-friendly APIs. AWS Cloud uses DynamoDB, a NoSQL database service that provides quick and predictable performance with seamless scalability. Chapter “9.1. AWS Amplify” in this document describes the reasons for making such choices in app architecture.

Based on the completed research, my primary frontend language will be React.js. It allows using web3.js vast libraries to request data with deployed smart contracts. The other reason for choosing React is that the company mainly develops applications using this language. After comparing frontend development languages, going through alternatives, and considering the company’s needs, React was selected. At the same time, an entire project structure will be handled nicely by having blockchain technologies and frontend components in a single project. This architecture is mainly used nowadays for decentralized projects (img. 0.4).

6.2. Lo-Fi design stage

First, I would like to mention that UI design is not my central sphere of interest in the IT world, but this is an unavoidable and essential part of this project. A certain number of things are required to start this stage, such as requirements specs, use cases, and company essential style guides (if available and required). As this project is developed for all devices: desktop, tablet, and mobiles, I must create the design accordingly. The lo-fidelity designed screens translate hi-fi design concepts into simplified and testable artifacts.

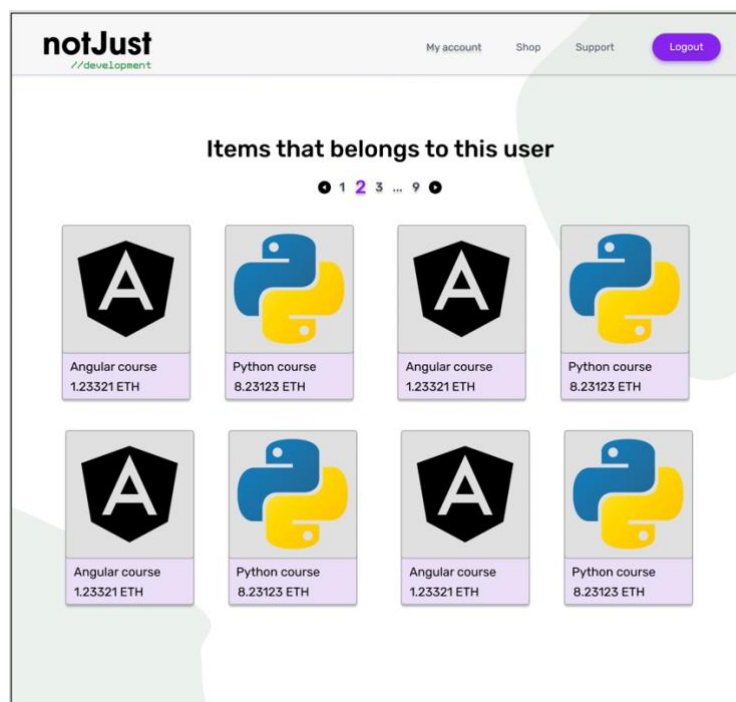


img. 0.5 – Lo-Fi design

I have tested the project Lo-Fi prototypes several times by going through the layout of the elements and flows for different stakeholders and finally achieved the result you can see above. I involved a contact person from the company looking from the course leader's point of view. And at the same time, I had an online meeting with the target group audience who tested the design from the course graduator's point of view.

6.3. Hi-Fi design stage

When I started to work on Hi-Fi design, I had already finalized and established Lo-Fi styling designs. I figured out a color pallet and general design for the future components. At this stage, I have tried to represent the product closely resembling the final version by improving the designs created at the Lo-Fi stage. One of the steps I took to achieve these results was analyzing and comparing existing websites that use a blockchain as their primary backend technology.



img. 0.6 – Followed design patterns

The image above (img. 0.6) displays all items a particular client has on a blockchain. They are accessible by auto-generated link through the app on the “My Account” tab from the navigation bar. This page contains a clear layout with eight items present on the screen. If the client owns more than eight items, the visitor can find others using a pagination component above the layout.

Overall, all other screens follow the same design principles: not overloading a page by showing many different items and components. I created the design to keep it simple and transparent for the user. The primary color pallet contains purple colors highlighting an important part of the pages. The user can easily find a relevant component when following a particular screen. The thesis attachment contains a .pdf file with the final version of the Hi-Fi prototypes.

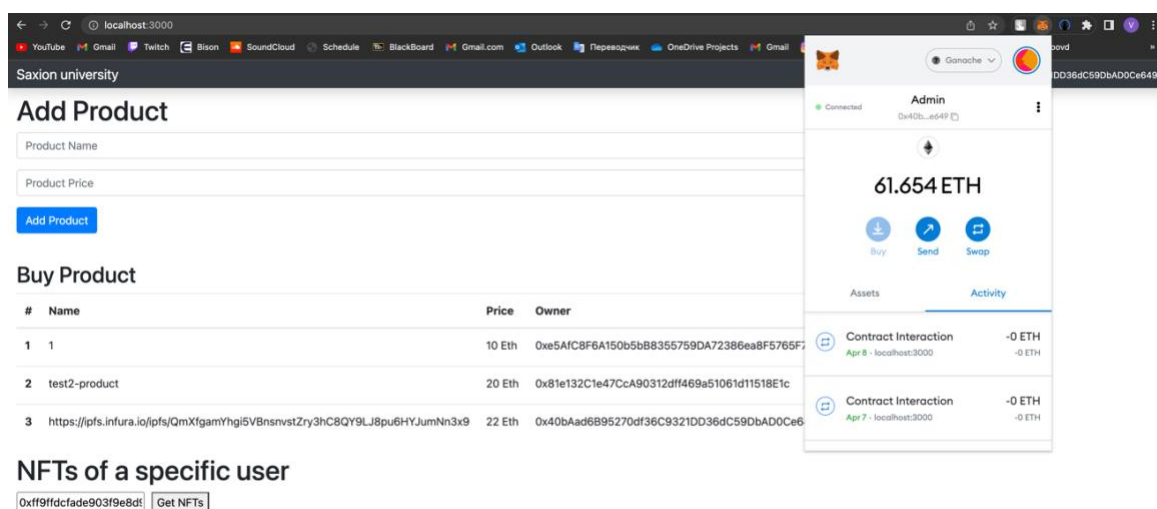
7. Proof of concept

I implemented the proof-of-concept application to show the possibility of building an application that will match client needs. It contains all functionalities that the system can combine to achieve the needs based on the requirements. The code of this application is structured in an easy way like the other development processes. It is because the primary purpose is to show functionalities and workability on the blockchain. Development mode is fully established on my side locally by using the Ganache application, which allows you to create a blockchain on your local network.

```
Contract: ImageStorage
deployment
  ✓ deploys successfully
  ✓ has a name (84ms)
images
  ✓ creates image
  ✓ list image (39ms)
```

img. 0.7 – Solidity tests

I am starting with the backend development. I had to understand how the chosen Solidity language works in more detail about data types, functions, and operators. Once I got acquainted with the language, I started implementing the functionalities until the system met the product requirements. During blockchain development, a developer must test new functionalities. The image above shows that the ImageStorage contract contains four tests to ensure the code is ready to deploy on a chain.



img. 0.8 – Proof of concept product

The image above (img.0.8) shows the final version of the implemented proof of concept application. Starting from the top, I have implemented functionalities in the column list. The first thing that I implemented was the authorization process. The main idea around authentication/authorization is to use the already existing crypto wallets of the clients. They must already have those wallets to make use of our product. For this purpose, I have used the “Metamask” browser extension, using which you can create/log in to your crypto wallet. And at the same time, this extension supports Ethereum transactions. My research papers can provide more detailed information on the chosen “Metamask” technology.

The proof-of-concept application on load tries to detect a web browser extension to make a proper connection. Once the connection is established, the current user can use all available functionalities. The possibility is to create a new product by using the name and price fields and submitting a form. This action will trigger a smart contract to add a new product to the list. For the actual product implementation, there is a requirement to add NFT objects that constrain additional fields. To process the image storage on the blockchain, I will use the IPFS server. I explained detailed information about it in research papers, but instead of storing an actual image on the blockchain, I will keep a path with the location of the image on IPFS. Each transaction on a chain goes under a certainly determined so-called gas fee. Reverting to the “Metamask” will act as a middleware between the frontend requests and the smart contract to pay for function execution.

Next to the listing, the client can fetch and see a list of already created items on a blockchain. The fetching data process exempts you from any gas-fee payments. Each item shows the name, price, and id of the owner. If the current user is not a creator, they will be able to buy a product by the price defined under the Price column. Once the client selects a consequence, the buying process follows similar steps. Metamask represents contract details with information on involved wallet addresses, gas fees, and final price.

Also, if an NFT owner account corresponds to the currently logged-in user, then an owner will be able to transfer this item at the end of the row. Company course leaders can assign NFTs to the course graduators using this feature. This process follows the same steps that involve gas-fee calculation without any charge to pay for the item itself.



img. 0.9 – NFTs related to the specific crypto wallet

An image above (img. 0.9) shows the implementation of searching an NFTs corresponding to the wallet id mentioned in the search input. The system can display all NFTs related to the particular crypto wallet id. I have used the “Moralis” open source servers to collect all NFTs that this wallet id retrieved before starting to use our services. It greatly benefits the user from the perspective that everything is in one place, and they don’t need to jump from platform to platform to see their items.

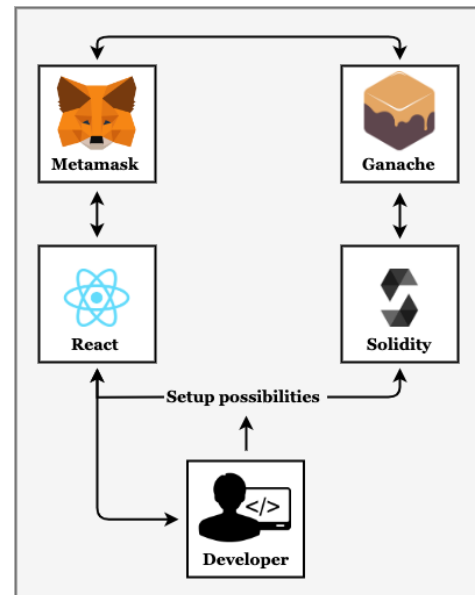
To conclude, I have delivered a proof-of-concept application with a small presentation where I went through each implemented functionality step.

8. Realization

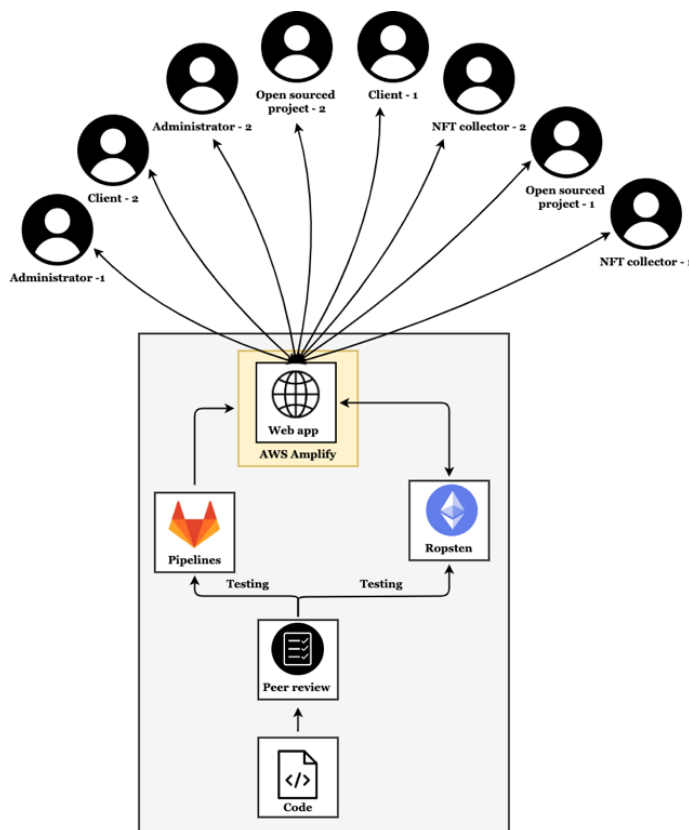
This chapter will contain detailed information on how a selected technology impacts the project. The images below (img. 1.0 & img. 1.1) show the software development lifecycle and deployment processes.

An image on the right (img. 1.0) shows all involved technologies to organize the development lifecycle. The developer can switch between blockchains, communicate with locals, or deployed smart contract.

React communicates through MetaMask to retrieve and exchange data with blockchain. The main purpose of solidity is to define the communication whenever the contract is called.



img. 1.0 – Development lifecycle



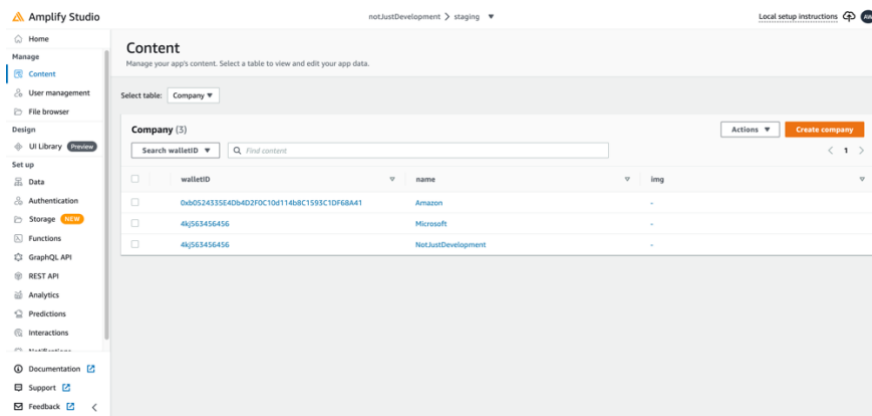
img. 1.1 – Feature deployment process

An image on the left (img 1.1) displays the flow of deploying new features in the product. If new frontend or backend features are released, they will be peer-reviewed first. The product will be updated and deployed with a more recent version if all pipeline stages are completed successfully. Finally, every involved stakeholder will be able to use the product for their needs. For example, the administrators will be able to certificate graduates digitally. Students will see those certificates on their accounts, etc.

8.1. AWS Amplify

Based on the complete investigations, I have chosen AWS Amplify instead of Azure or Linux services for centralized data management. The reason is that Amazon is a leader in developing and selling its worldwide well-known SaaS services. There were other criteria based on which I followed AWS Amplify. Investigation comparison and results are listed in the attached files.

This project makes use of AWS's built-in tools and features. The primary purpose is to give a possibility to store companies administrators' crypto wallets id's on blockchain to provide them with appropriate permissions on the frontend. This way, the product owner can insert a new trusted administrator in the system and grant a possibility to issue a digital certificate using the web app. Database stores company object that holds an id, walletID of the administrator, company name, and image. All fields apart from the image are mandatory while creating a new company object.



img. 1.2 – Amplify Studio model data

The second reason for using centralized storage within this project is to show all company's information on NFTs cards. Every item on sale will represent defined fields such as the name, description, and price of an NFT. By default, an original owner of an NFT is the administrator of the trusted company. It will be recorded in the Amplify Studio content tab. After the subsequent sales within the platform, buyers will override the original administrator owner wallet id. If the product belongs to the certificate group and has been sold several times to different users, it still must show which company initially issued this sticker.

When the client opens an item from the market, the system will go through the defined trusted administrators and companies where they are working. If opened item creator id corresponds to the issuer id from AWS data storage, the current item will include the company details that issued this sticker.

Thirdly, the usage of AWS Amplify can be seen within a product to determine whether an item corresponds to the market. Objects created by trusted persons can be either sellable or not. If the creator defined an object type as “On sale” with the following price of more than zero, AWS Amplify would store an id and price of this item on the centralized data storage. If an item is not sellable, there would be an option to put it on the sale in the future by setting a price on a different tab. The shop screen fetches objects from blockchain and centralized data storage to display only items whose ids correspond to the ids and prices stored within Amplify.

8.2. Solidity

Within this project, Solidity is a primary backend-blockchain development language. Based on the research stage, I have decided to develop smart contracts by writing a code with solidity over the other candidates. The detailed comparison and decision-making are described in the research papers.

The backend structure contains only one model representing an NFT with the following fields: id, price, url, name, imageType, description, creatorId, and owner. Those fields for an NFT make it possible to create the following functions of creating, purchasing, and issuing that cover requirements scope needs for this product. One important thing to clarify about solidity is that there are only three types of functions: “call” to fetch, “send” to post data, and “transfer” to pass a defined amount of money to the specific wallet. The transfer function includes a defined callback function, or the transfer call will throw an error. The call one can be used to call public and external functions on contracts and does not require any transaction fees to obtain data. The send works similarly to transferring the call and has a gas limit of 2300 gas (Mohammed, 2021). One important detail about the backend is that it stores a wallet id of a company. Buying operations on our platform will include a 3.5% fee from the product price to the company itself.

The main thing that I have considered while developing new functionalities is testing. Tests must cover all new features and bug fixes in existing code. Smart contracts are stored on the blockchain, which means they are decentralized. Once the backend is deployed, every further change in the code will cost a certain amount of money based on the blockchain. So, ensuring that all functionalities are tested well before deploying is essential. For this reason, the development stage is entirely done through the sample blockchain environment created locally on my laptop.

8.3. React

I have chosen one of the most popular new web frameworks among the frontend communities for frontend development, used by many innovative companies. It is a JavaScript library for creating user interfaces (Academy, 2016). As I have already mentioned in Chapter 6.1. Architecture design, the company preferred to make use of React as a frontend language.

This framework allows communication with the Ethereum blockchain. To enable smart contract interaction within the application, I installed a web3.js library. The web3 JavaScript library interacts with the Ethereum blockchain. It can retrieve user accounts, send transactions, interact with smart contracts, etc.

As shown in the diagram in Chapter 7.1. “Architecture design” user interfaces have defined functions to execute on the smart contract. Besides communicating to the remote Ethereum nodes, HTTP implementation enables interaction with local nodes.

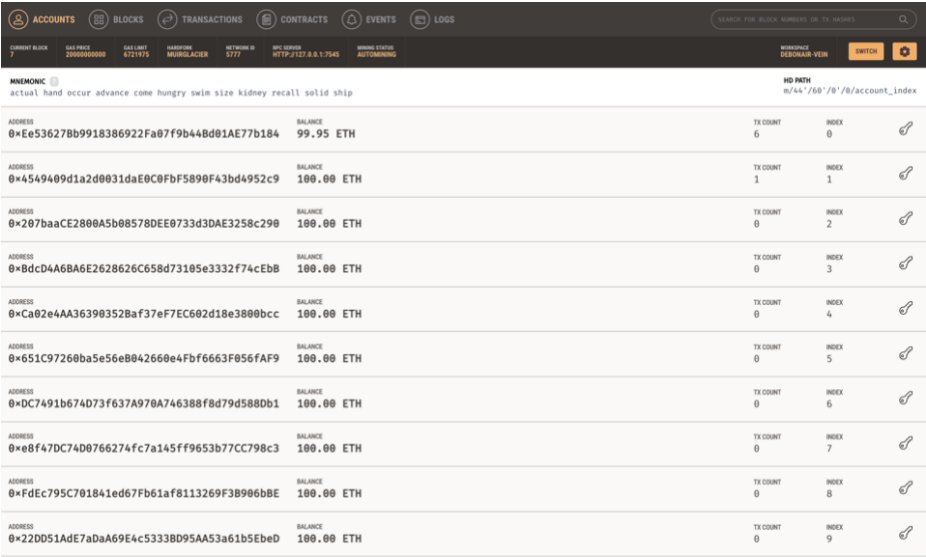
The usage of React.js within this project follows an “Atomic design” pattern. The primary idea is to design systems with five fundamental blocks that promote modularity and scalability when combined.

In the beginning, the development starts with “Atoms.” It is the fundamental building block of matter, such as buttons, input, etc., which is not helpful. Secondly, “Molecules” groups atoms together by combining controls and inputs with building functionality. Those “Molecules” forms “Organisms” by connecting them that make up a distinct section of an interface. Then templates come up by grouping an “Organism” to include a page – where clients can see a final design. Finally, an “Ecosystem” that views different templates renders. To conclude, creating multiple “ecosystems” into a single environment is called – the application. (Wong, 2017)

8.4. Ganache

The development stage is an exciting and vital part of this project. Since the main requirements functionalities will be completed on the blockchain, it may turn costly for the company. That is where the Ganache comes to solve this problem. It is a personal blockchain for rapid Ethereum distributed application development. It can be used across the entire development cycle. It helps me develop, deploy, and test my backend in a safe and deterministic environment.

The choice of using Ganache was straightforward. It was the first technology I found and investigated thoroughly. I did not make a further investigation across other candidates because it covers all development needs.



ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS
MEMORIC actual hand occur advance come hungry swim size kidney recall solid ship					
ADDRESS 0xE536278b9918386922Fa07f9b448d01AE77b184 BALANCE 99.95 ETH TX COUNT 6 INDEX 0					
ADDRESS 0x4549409d1a2d031daE0C0FbF5890F43bd4952c9 BALANCE 100.00 ETH TX COUNT 1 INDEX 1					
ADDRESS 0x207baaCE2800A5b08578DEE0733d3DAE3258c290 BALANCE 100.00 ETH TX COUNT 0 INDEX 2					
ADDRESS 0xBdcD4A6BA6E2628626C658d73105e3332f74cEbb BALANCE 100.00 ETH TX COUNT 0 INDEX 3					
ADDRESS 0xCa02e4AA36390352BaF37eF7EC602d18e3800bcc BALANCE 100.00 ETH TX COUNT 0 INDEX 4					
ADDRESS 0x651C97260ba5e56e042660e4FbF6663F056FAF9 BALANCE 100.00 ETH TX COUNT 0 INDEX 5					
ADDRESS 0xDC7491b674073F637A970A746388f8d79d588db1 BALANCE 100.00 ETH TX COUNT 0 INDEX 6					
ADDRESS 0xe8f47DC74D0766274fc7a145ff9653b77CC798c3 BALANCE 100.00 ETH TX COUNT 0 INDEX 7					
ADDRESS 0xFdEc795C701841ed67Fb61af8113269F3B906bbe BALANCE 100.00 ETH TX COUNT 0 INDEX 8					
ADDRESS 0x22D051AdE7aDa69E4c5333BD95AA53a61b5EbeD BALANCE 100.00 ETH TX COUNT 0 INDEX 9					

img. 1.3 – Local blockchain network account

The image above (img 1.3) shows the local blockchain information after establishing it on my machine. It simulates ten user crypto wallets with all relevant information usually presented on an actual blockchain. Next to the account wallet id gives a hundred simulated ETH, which can be used within this local environment to test backend functionalities.

8.5. Metamask

Based on my findings, I chose MetaMask as a digital wallet among other candidates. Nowadays, it is the most well-known digital wallet, and alternatives to that usually do not have that brought functionalities. I wrote conclusions and comparisons in my research papers.

Metamask provides an essential utility for the Ethereum blockchain and is responsible for authorization processes within the app (Metamask, 2020). The platform has an option to log in with Metamask. The correspondent notification will be shown if the client does not have a Metamask extension or the browser does not support such libraries. Once the user is logged in, access to their Ethereum wallet is granted through the web extension. At this point, every user/administrator can easily buy, sell, or issue an NFT covering gas fees with their ETH correspondingly.

Using Metamask, users can use wallets to store Ethereum-compatible and non-fungible tokens (NFTs) across multiple blockchains, including the Ethereum main blockchain, Binance BNB blockchain, Polygon, Avalanche, and the test networks. (CORTEZ, 2022)

8.6. Ropsten

Several services can be used as a test network for EVM-based applications. Those are Ropsten, Kovan, Rinkbery and Gorli. They are the same and give similar possibilities explained about Ropsten in this chapter. Every candidate is a test network, a Proof-of-Authority test for Ethereum. So, I made no comparisons between them in the research document. In this chapter, I will go through the usage of Ropsten within the platform.

At the beginning of the project, we did not plan deployment to be a part of the graduation assignment. But, at the end of the project, I had enough time to spend on it. Deploying a developed solution on a Mainnet will cost a company some money, which is not what they wanted. This stage must be accomplished only when a professional tester tests the entire product. The project contains test cases for both frontend and backend, but it is not enough to deploy a project on the market. The solution that I found is to make use of a test network. This way, the company can see how the released product will look without paying any money. And when it is tested well, the company can easily switch from Ropsten to a regular Mainnet network without any difficulties.

The Ropsten is an Ethereum-based test network that allows blockchain development testing before the smart contract deployment (AnyBlock, 2021). On the Ropsten network, the Ethereum money “Ether” has no real value. Any developer can access those funds for free. The network provides several links where people with crypto wallets who use the Ropsten network can request some ETH. Those websites give from 0.3 ETH up to 1 ETH in 24 hours. As I have found, mainly request checks the IP address so the wallet can get several payments daily by switching the requestor IP. This can be accomplished by using VPN.

The screenshot displays the Etherscan interface for the Ropsten Testnet Network. At the top, the Etherscan logo and network name are visible. A search bar and navigation links (Home, Blockchain, Tokens, Misc) are present. The main section shows a contract overview for the address 0x9f1b3daa348f855239E7106F8c7f824666FD78f3. The contract's balance is 0 Ether. Below this, a table lists the latest 7 transactions, including transaction hashes, methods, block numbers, ages, from/to addresses, values, and fees. The transactions show a series of calls to the contract, with the final one being the contract creation.

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x474b983e81dc291a5c...	0x14e5ce32	12239988	21 hrs 50 mins ago	0xc4a79123923bcf9b40f...	0x9f1b3daa348f855239e...	0 Ether	0.004653613820
0xc95d70fcd3c6d440c2f...	0x14e5ce32	12234298	2 days 21 hrs ago	0xc4a79123923bcf9b40f...	0x9f1b3daa348f855239e...	0 Ether	0.012221931483
0xe15d06616935e8e1f0...	0x705a88a	12232948	3 days 17 hrs ago	0xc4a79123923bcf9b40f...	0x9f1b3daa348f855239e...	0 Ether	0.001578490244
0xc02ac1a656840910c...	0x14e5ce32	12232928	3 days 17 hrs ago	0xc4a79123923bcf9b40f...	0x9f1b3daa348f855239e...	0 Ether	0.01348049316
0x11ed03573c5a69c120f...	0x14e5ce32	12232924	3 days 17 hrs ago	0xc4a79123923bcf9b40f...	0x9f1b3daa348f855239e...	0 Ether	0.014717334849
0xea30e43ee126beffdb3...	0x14e5ce32	12232914	3 days 17 hrs ago	0xc4a79123923bcf9b40f...	0x9f1b3daa348f855239e...	0 Ether	0.011496501145
0x86cd2e8516183bc93ff...	0x60806040	12232754	3 days 19 hrs ago	0xc4a79123923bcf9b40f...	Contract Creation	0 Ether	0.048095698515

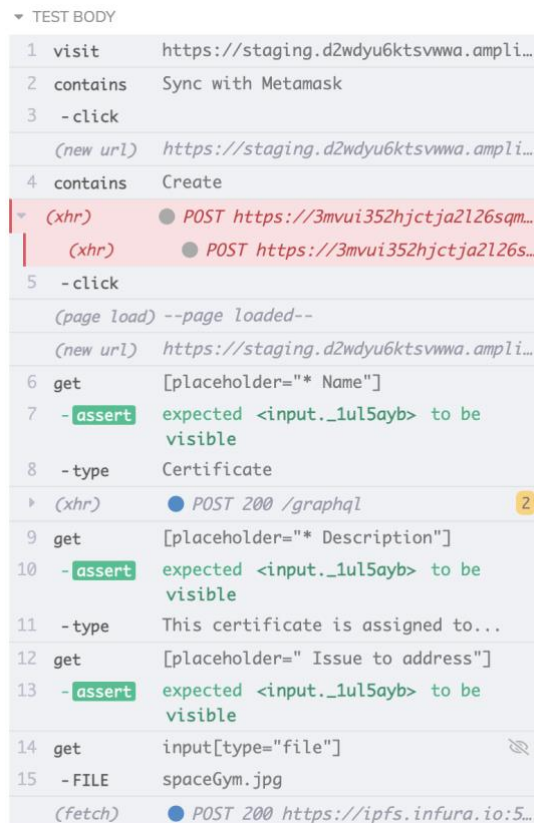
img. 1.4 – Deployed contract on Ropsten

As mentioned in this chapter, Ropsten is an Ethereum-based network observing similar deployment and maintenance steps. For example, the image above displays all transactions executed on the blockchain in ropsten testnet explorers. It has a hash, method, block, age, wallet details, values, and fee information about every transaction. The interface is similar to the Ethereum network explorer. During testing, it can access it is easily accessible by adding a “ropsten.” to the beginning of the official “etherscan.io” address.

Overall, usage of the Ropsten test net network helps testers work with the deployed product on the blockchain. But the developer does not have to pay all gas fees to mint the smart contract on the Mainnet. Once all tests are passed successfully, the developer can request funds from the company to cover deployment gas fees. After testing a product on a test net, the company ensures that the deployed product encompasses all requirements and works as expected.

8.7. Frontend testing

The frontend technology which is used in this project is cypress.io. It is a user-friendly test automation tool for UI testing, regression suites, integration, and unit testing. In this project's scope, cypress tests the product functionalities of a single unit and aggregates results from various parts of the screens.



TEST BODY	
1	visit https://staging.d2wdyu6ktsvwwa.ampli...
2	contains Sync with Metamask
3	-click
	(new url) https://staging.d2wdyu6ktsvwwa.ampli...
4	contains Create
5	(xhr) POST https://3mvui352hjtja2l26sqm...
	(xhr) POST https://3mvui352hjtja2l26s...
5	-click
	(page load) --page loaded--
	(new url) https://staging.d2wdyu6ktsvwwa.ampli...
6	get [placeholder="* Name"]
7	-assert expected <input._1ul5ayb> to be visible
8	-type Certificate
	(xhr) POST 200 /graphql 2
9	get [placeholder="* Description"]
10	-assert expected <input._1ul5ayb> to be visible
11	-type This certificate is assigned to...
12	get [placeholder=" Issue to address"]
13	-assert expected <input._1ul5ayb> to be visible
14	get input[type="file"]
15	- FILE spaceGym.jpg
	(fetch) POST 200 https://ipfs.infura.io:5...

The image on the left (img. 1.5) displays a result of a single test file executed using cypress. Generally, it uses an arrange, acts, and assert steps to make a proper test:

1. It puts the components and operations required for a test case.
2. It acts as defined in a test file.
3. It compares the resulting outcome of the test with the expected established behavior.

To cover an entire project with tests, every screen has its test file where all testing components and their behavior are defined.

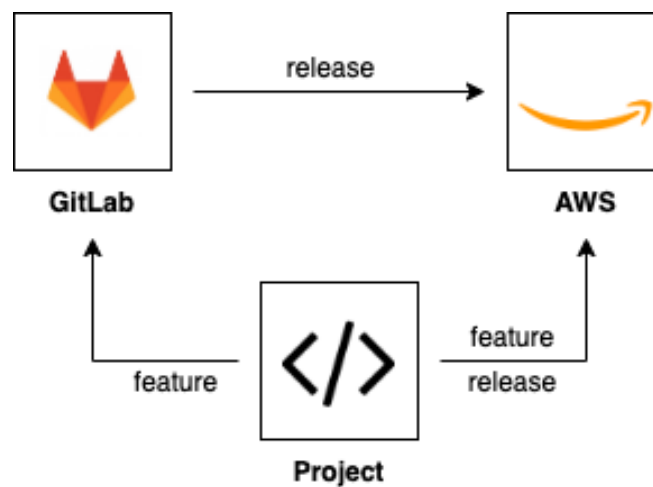
img. 1.5 – Cypress test steps

Tests will wrap only mandatory functionalities and components with tests for the final product version. Generally, the frontend testing contains a fully established environment that can increase with more complex test cases.

8.8. Frontend deployment

The frontend application is built using React.js and Web3 libraries to communicate with the blockchain. The app handles the user interface according to the created version of the High-Fi designs combined with blockchain data exchange.

The deployment stage is accomplished by using AWS hosting services. To deploy a product, I have created a separate project that hosts the remote's source code. It generates a random domain name using which clients can access the deployed product.



img. 1.6 – Deployment possibilities

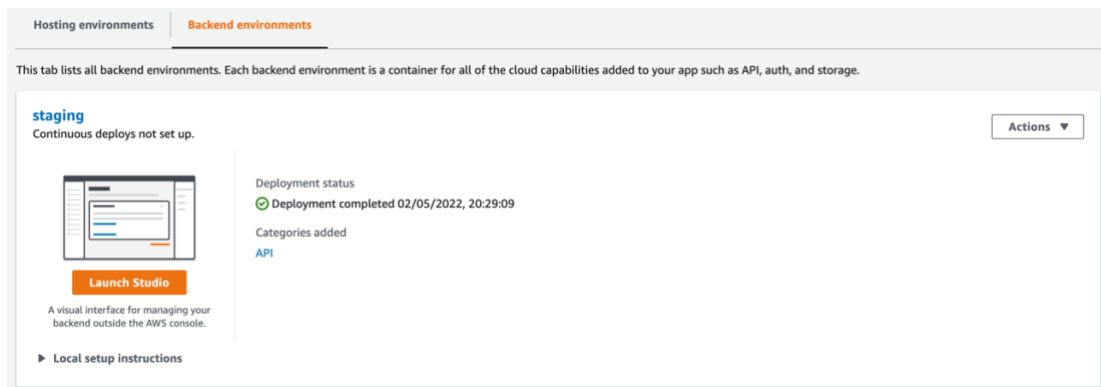
The diagram above (img. 1.6) shows two possible flows of deploying a product to the remote. Firstly, the AWS project can be linked with GitLab. By establishing a proper connection between the chosen branch on the git, AWS will keep track of new updates and automatically redeploy the product with the latest version. Secondly, developers can publish updates from the terminal. This product uses the second version of deploying updates during the development stage of the product. After the development stage, the best option would be to switch it and use the first opportunity with automatic releases. As well as the domain that can be paid and changed from the default generated one.

8.9. Backend testing

Testing smart contract functionalities is an essential part of the project. As mentioned in chapter 8.2. Solidity tests wrap up every new functionality that the developer created. The main reason is to avoid a failure of functionality to be deployed on the blockchain. It may result in a significant financial issue for the company because of the gas fees it takes to publish changes.

8.10. Backend deployment

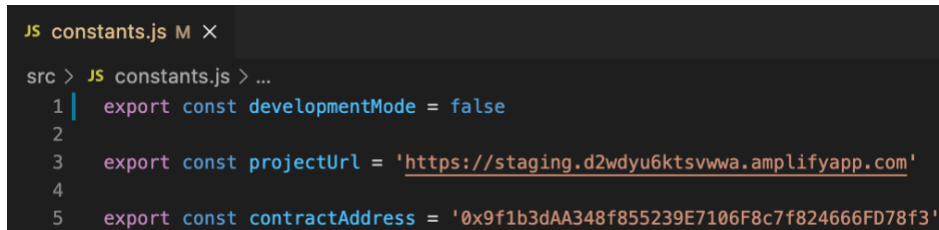
The backend that related to the deployed functionalities mentioned in chapter 8.1. AWS Amplify is stored on AWS servers. It uses centralized storage and API communication when receiving and sending data. Generally, it acts as centralized storage and only contains designed data models. The image below displays that I deployed both frontend and backend applications on a single server. The main benefit of using AWS is that it is free and has a self-explanatory and understandable user interface. At the same time, it supports API calls using GraphQL query language for CRUD operations.



img. 1.7 – AWS Amplify backend container

8.11. Setup possibilities

The image below (img. 1.8) shows a file created in the project's root directory. I have configured this file with the essential data used in the product to function correctly. Using this setting, developers can easily switch an entire application into development mode and use it with the local blockchain. This operator only affects the way how the frontend connects to the blockchain. All screens and components' behavior do not depend on this Boolean variable.

A screenshot of a code editor window titled 'JS constants.js M X'. The editor shows the following code:

```
src > JS constants.js > ...  
1 | export const developmentMode = false  
2  
3 | export const projectUrl = 'https://staging.d2wdyu6ktsvwwa.amplifyapp.com'  
4  
5 | export const contractAddress = '0x9f1b3dAA348f855239E7106F8c7f824666FD78f3'
```

img. 1.8 – Setup possibilities

Next to it, the frontend project's deployed url is present in the file. And at the last position, the address of the deployed smart contract is mentioned. All fields must correspond to the data before deployment or during development. Otherwise, an entire system will not work as expected.

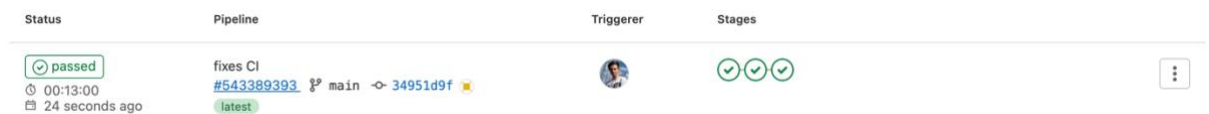
8.12. Pipelines

I implemented the pipelines on a web-based git repository where the product's source code is stored. In the scope of this product, pipelines are executed in three steps.

Firstly, it updates and loads all used npm packages of the product. It verifies that packages are workable and do not interact with each other producing incompatible version errors.

The next step verifies that the code style corresponds to the defined style rules. It checks all javascript-based files and runs a check to verify code styling patterns.

Finally, the last step deploys a created blockchain backend. It does it on a test network described in Chapter 8.6 Ropsten. Once the smart contract is deployed, the second part of the tests begins. It verifies that all blockchain functions covering requirements from a backend point of view are valid and workable.



img. 1.9 – GitLab pipeline

The image above (img 1.9) displays a pipeline with all successfully passed tests. The new features did not affect the product quality and functionalities integrated before.

9. Validation

To validate all used technologies and the completeness of the product, the following aspect is going to be explained in this chapter:

- Completeness: auto, manual tests, test cases
- Robust – deployment
- High quality: peer reviews, linters, prettier and pipelines

9.1. Tests

I tested all requirements with completeness to prove that they function correctly according to the initially defined requirements. The automation testing is explained in chapter 8.7. Frontend testing and 8.9. Backend testing. For both frontend and backend, appropriate testing environments were setup. The backend was automated and attached to the pipeline build process on Git. The testing must be done manually with the frontend because the MetaMask authorization process requires a manual login step. Due to the limited time for this project, this would be my first recommendation on product improvement for the future.

I completed the manual testing at the end of the project. It tests all MUST requirements by following a defined structure mentioned in the document attached to the Thesis, the “Manual_Testing” file. It also has defined test cases through which the system tested the main requirements. The results were positive, proving the workability of developed functionalities covering company needs.

9.2. Robust

From the deployment point of view, the product is available by following a remote URL. All necessary parts of the project were set to be working remotely. I established the communications between the deployed parts of the product. As mentioned in Chapter 8.11. “Setup possibilities” developers can configure the product from the root directory. It can be quickly adapted to work in the development stage and be switched and deployed to be available on the web.

9.3. High quality

As mentioned in Chapter 4.1. Project Management, the project follows SCRUM as the development methodology used in software development. We established a peer review session with the company supervisor to ensure software quality. More detailed information can be found in Chapter 4.4. DevOps Platform.

The entire development process is established under specific code rule standards. We agreed with the company to use a static code analysis tool – eslint. At the same time, prettier was used as the code formatting tool. More detailed information about those technologies can be found in Chapter 4.5, Quality assurance.

Finally, automation tests take place before deployment. This step is reached at the end. Once all processes of the first two chapters of this paragraph are accomplished. The pipeline on GitLab ensures package versioning and project workability and runs backend testing. For the frontend, testing must be manually launched using an established testing environment. The reason for that is explained in Chapter 9.1. Testing.

10. Conclusion

I want to start my conclusion with the recommendations for the company that I have. First, I would suggest doing a proper product test. It is up to the company to deliver high-quality software to the client, but testers must appropriately test the product before the official release. I recommend hiring a testers team that will go through each functionality and try to spot bugs the system possibly has. The main reason for that is because the scope of tasks has expanded since the beginning of the development. By the end, I had integrated all established requirements for this product and covered them with automated testing.

Secondly, I know that the company holds one unused domain through which customers can access the project. A client will see a company-related domain name instead of an automatically generated one from AWS.

I am trying to be abstract while reviewing the work that I did. I have mentioned several improvement points for the project in the thesis. The software follows professional working flows and techniques studied at Saxion and on my own. It will be easy for the company to keep working and increasing software functionalities. At the same time, maintenance is carried out well and can expand in the future with more refined tests. Finally, the last point to clarify, the web application is adapted to work on all devices.

Overall, I found this project interesting for me. As I have already mentioned to the teachers, I am very thankful for Saxion giving students such possibilities to experiment with different development technologies while studying. Blockchain development, for me, was an exciting and challenging experience. I want to emphasize that everything is connected very logically and understandable. Having a Web2 knowledge base studied at Saxion helped me get into the project faster. As per my thoughts, I have grown up as a software developer specialist very well after this project.

11. Bibliography

- Mohammed, Z. (2021, december 31). *Solidity — transfer vs send vs call function* . Retrieved April 2022, from Medium: <https://medium.com/coinmonks/solidity-transfer-vs-send-vs-call-function-64c92cfc878a>
- Pages, G. (2021). *Secure ether transfer*. Retrieved April 2022, from GitHub: https://fravoll.github.io/solidity-patterns/secure_ether_transfer.html#:~:text=In%20the%20early%20days%20of,the%20function%20is%20called%20on
- Tiwari, A. K. (2022, March 3). *What is Ganache?* Retrieved April 2022, from Medium: <https://medium.com/coinmonks/what-is-ganache-how-to-use-it-for-blockchain-project-development-85eff9f1f728#:~:text=Ganache%20is%20a%20personal%20blockchain,is%20your%20local%20blockchain%20simulator>
- Metamask. (2020). Retrieved April 2022, from Metamask: <https://metamask.io/>
- AnyBlock. (2021). *Main Ethereum Proof-of-Work Testnet*. Retrieved April 2022, from AnyBlock: <https://www.anyblockanalytics.com/networks/ethereum/ropsten/>
- Academy, C. (2016, June 10). *Top 32 Sites Built With ReactJS*. Retrieved April 2022, from Medium: <https://medium.com/@coderacademy/32-sites-built-with-reactjs-172e3a4bed81#:~:text=ReactJS%20is%20one%20of%20the%20most%20popular%20new%20web%20frameworks,function%20hooks%20to%20output%20HTML>
- Kenneth, H. (2019, March 19). *Web3 .js Ethereum Javascript API*. Retrieved April 2022, from Medium: <https://medium.com/coinmonks/web3-js-ethereum-javascript-api-72f7b22e2f0a#:~:text=Ethereum%20Javascript%20API-,web3.,with%20smart%20contracts%2C%20and%20more>
- CORTEZ, J. (2022, 05 14). *MetaMask Cryptocurrency Wallet Review*. Retrieved from Investopedia : <https://www.investopedia.com/metamask-cryptocurrency-wallet-review-5235562#:~:text=types%20of%20transactions-,Supports%20all%20ERC%2D%20tokens%2C%20NFTs%2C%20and%20multiple%20blockchains,Avalanche%2C%20and%20other%20test%20networks>
- Wong, J. (2017, 12 11). *Atomic Design Pattern: How to structure your React application*. Retrieved from Medium: <https://medium.com/@janelle.wg/atomic-design-pattern-how-to-structure-your-react-application-2bb4d9ca5f97>
- Wikipedia. (2022, 02 15). *Notion*. Retrieved from wikipedia.org: [https://ru.wikipedia.org/wiki/Notion_\(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5\)](https://ru.wikipedia.org/wiki/Notion_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5))