PaySmart Final Report:

Bunz Hackathon

Must include:

1. Final Dapp URL (staging environment) deployed on any testnet + github repo

   URL: https//paysmart.vercel.com/
   github Repo: https://github.com/holyaustin/paysmart

2. *DApp description*: what type of application it is (use case), which Bunzz modules were used and what was the intended functionality

   This project was made using several technologies. The front-end was designed using a server-side-rendering javascript tech known as NextJS. the latest version of Next was used because of how fast it was to build the project.
   A user can creates a new pool by clicking either of the following
   1. Add Pool (Native) MATIC or
   2. Add Pool (ERC20) PST Token Mint PST Token.

   When you create a pool a pool with the PST token, you have to approve the contarct to spend the PST token on your behalf befeore tramsfering token to the smart contarct.
   Once deal is completed, the agent releases token to the reciever and get its percentage.
   If you dont have the PaySmart (PST) Token, you can mint it by clicking on Mint PST Token.
   The entire project demo was deployed to Vercel

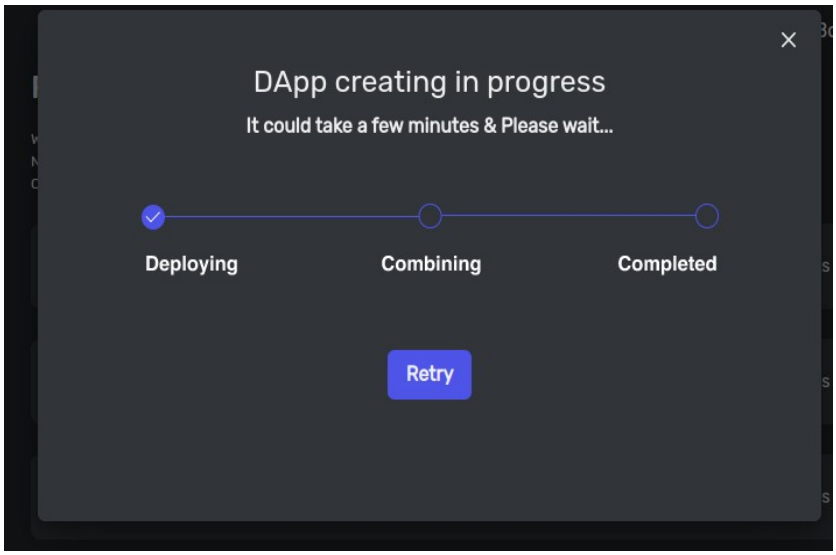
3. Did you find in Bunzz all the smart contracts needed to give your DApp the expected functionality? Or did you have to import other external contracts to complement the functionality available through the modules offered in Bunzz?

   Yes, Bunzz Provided the erc20 and the escrew smart contact that I used for the Paysmart Dapp. Based on my based, it did not require external import.

4. Did you experience any trouble finding, selecting, combining, understanding specifications and/or deploying your smart contracts modules in Bunzz?

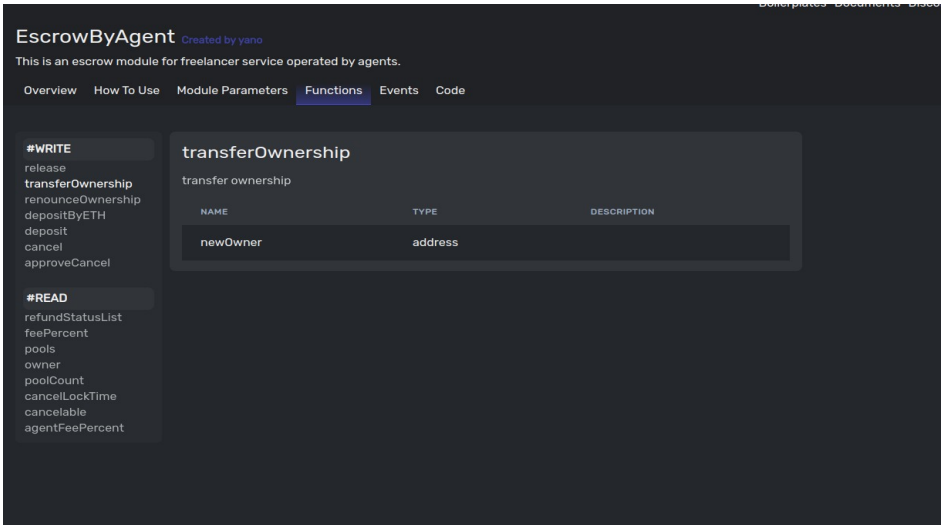
   Yes I had few issue when selecting nad combining modules. The erc20 module could not combine with the escrew module. I had to deploy them as individual modules.

When deploying escrow and erc20, it gets stuck on combining.



5. Did you experience any error while passing arguments or deploying a module along your usage of Bunzz? Please describe and ideally attach screenshots.

Yes, I had a challenge with the escrow module. I had to inspect escrow smart contract to understand how to implement the depositByETH function on the frontend. It had two input parameters without the amount parameter. This was implemented with the msg.value on the smart contract but there was no clear description about it on the documentation. I had to figure it myself.




```
function depositByETH(
  address _recipient,
  address _agent
) external payable override returns (uint256) {
  require(msg.value > 0, "amount invalid");
  require(
```

```
    _recipient != address(0x0) && _agent != address(0x0),
    "address invalid"
  );
  return _deposit(address(0x0), msg.sender, _recipient, _agent, msg.value);
 }
```

Msg.value wasnt clear to me until I have to inspect the smart contract. It should be clear from documentation or a variable amount should be created the way it was done in the deposit() function.

6. Once you deployed your smart contracts modules. How did you find the process of connecting them with your frontend application? Was the Bunzz documentation helpful and enough? If not, what should be added to make it clearer and easier to understand.

   Connecting the deployed contract to the frontend, it was pretty easy using ethers.js to interact with the smart contract. Except for the fact that I got mislead by two input variable for the function depositETH without a clear explanation on how to get the amount into the contract.

7. User Interface:
   * Describe the framework you used and why you chose it.
   NextJs was the front end framework used. By using Next. js, you're able to do things like server-side rendering, which means the server reads through code and tells the browser what to display on the page. Server-side rendering creates faster, more versatile web applications.

   * Did you experience any difficulties when integrating smart contracts modules's functions into your buttons, forms, wallet connection, notifications, etc.? If yes, please specify and describe errors that you experienced.
   No Difficulty

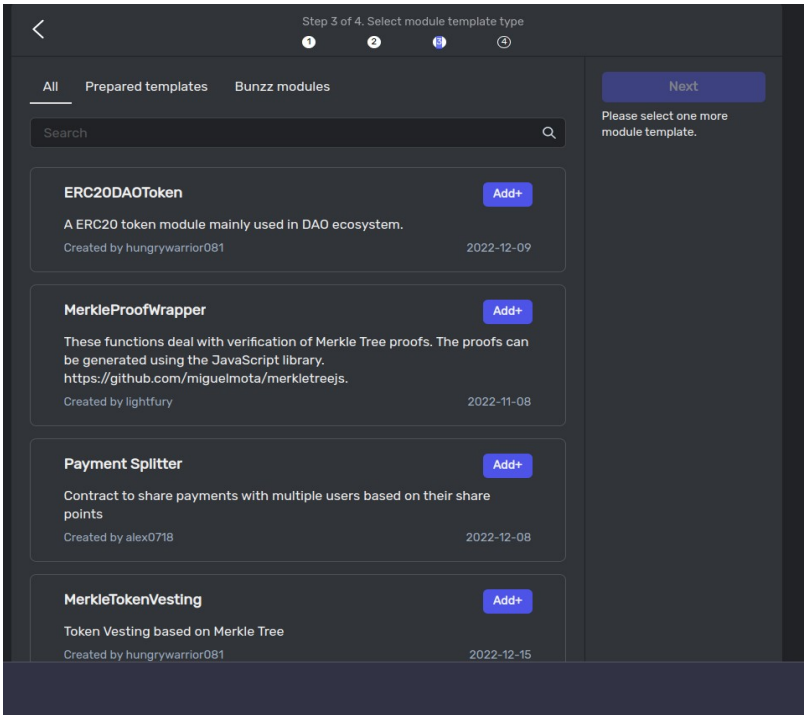
   * Do you consider there is something missing related to infrastructure or UI features that Bunzz offers for you or any other developer could be able to build a full stack DApp from beginning to end?

   Comprehensive documentation will project examples will help onbaoard new users on how to build a fullstack Dapp using Bunzz.

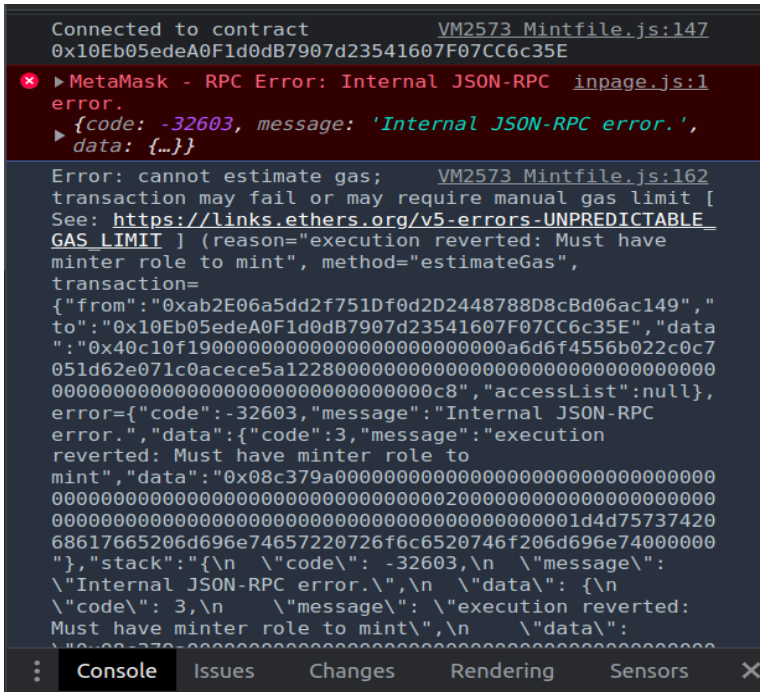
Others

1. One cannot select multiple Templates / Modules to use at once.

Escrow and Splitter was not possible. I want to test an idea with both, I couldn't.



2. Got this error when integrating a function from the smart contract. “TypeError: Cannot read properties of undefined (reading 'map')”.



Cannot estimate Gas Error, I dont know if it is specific to Polygon Mumbai or from the Bunzz Module.
Workaround was adding a high gas Manually.