

## **Data Dashboard Developer: Project Report – Sam Yard**

Overall, the goal of this user-friendly dashboard is to provide easy access to sustainability data from the blockchain, offer meaningful insights and recommendations, ensure data integrity and authenticity through verification mechanisms, and promote waste reduction strategies among faculty, students, and other users in a visually appealing and user-friendly manner.

### **Dashboard Design Outline:**

#### **Backend:**

- **Authentication**: The dashboard has a user-friendly authentication process that allows faculty, students, and other users to securely access the dashboard. Involving email-based or sign-up / login options. Once authenticated, users should be authorized based on their roles and permissions to ensure data confidentiality and integrity.
- **Data Aggregation**: The dashboard intakes sustainability data from the blockchain at regular intervals (e.g., daily) and aggregates it to display meaningful insights. This can be done by combining related data into a pandas dataframe so that it can be analyzed easily. For example, computing averages, standard deviations, as well as creating visualizations.
- **Verification of Data Integrity**: Providing a mechanism for users to verify that the displayed aggregate data results are based on the data from the blockchain is imperative. This could involve displaying the digital signature and timestamp of the data that is collected.
- **System Design**: I chose Solidity to create the smart contract because it is a high-level language, compared to Rust. The smart contract was compiled and deployed through the Remix IDE. The addition of a smart contract to the system is crucial in gaining user trust. The contract bytecode is stored on the blockchain and makes sure that the data is verified and cannot be tampered with from unauthorized sources.

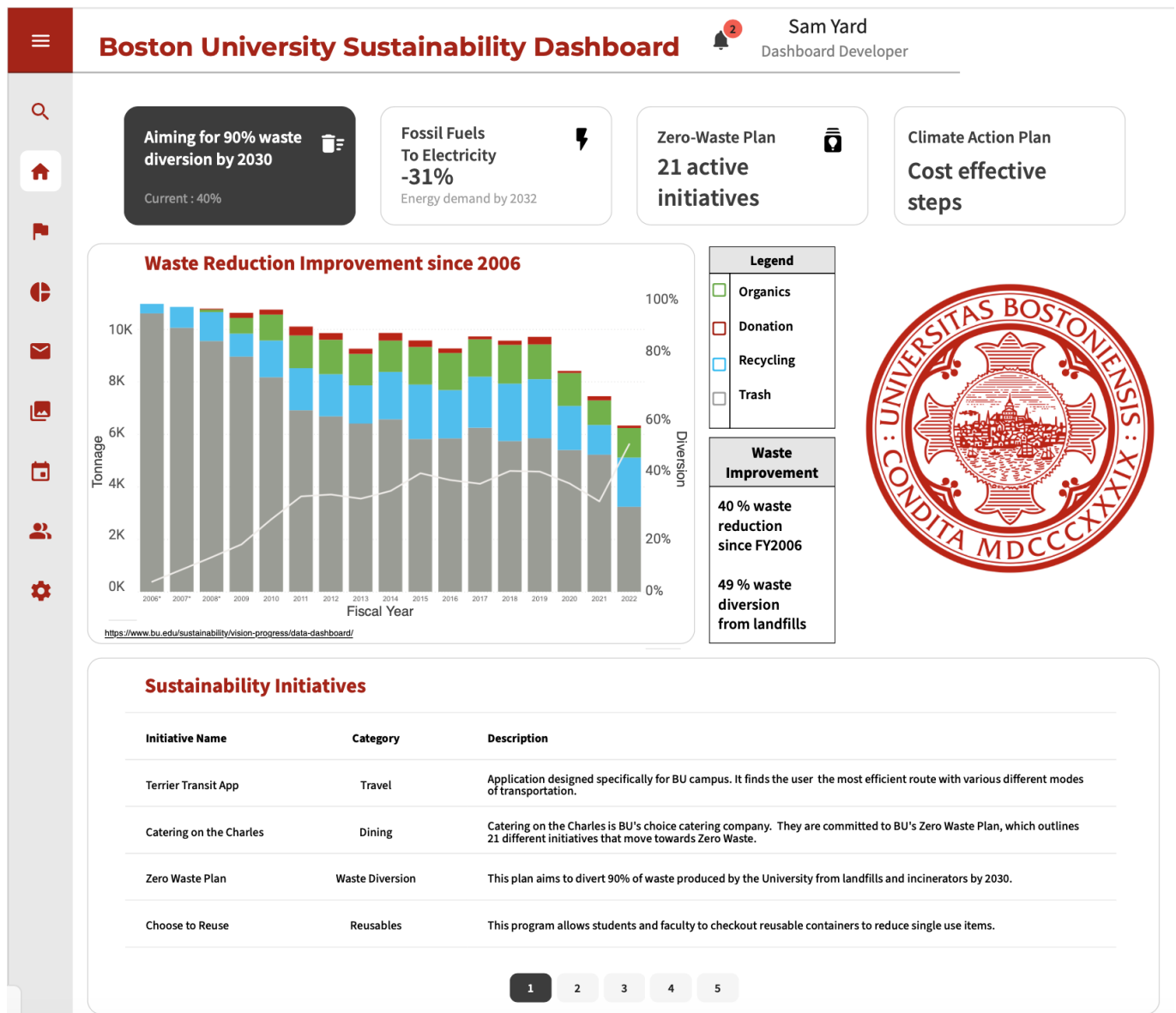
#### **Frontend:**

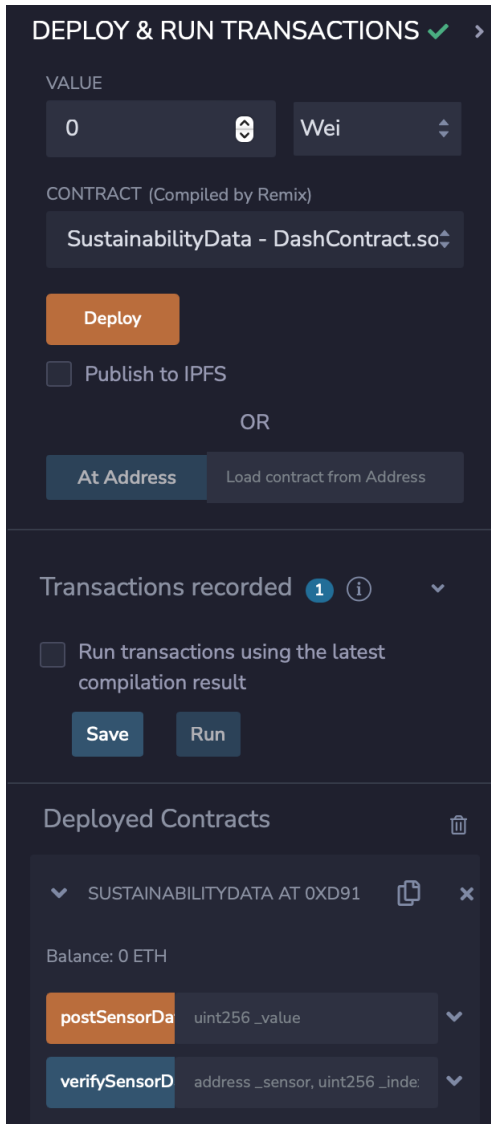
- **Insights and Recommendations**: Use data analytics to derive insights from the aggregated data and provide recommendations on waste reduction strategies. For example, it could highlight areas where sustainability goals are not being met, suggest actions to reduce waste generation or promote recycling, and provide tips for optimizing resource usage.
- **User-Friendly Interface**: A clean user interface that is optimized for web and mobile platforms. It should be responsive, visually appealing, and easy to navigate, with clear labels and instructions. The dashboard should also be designed to work seamlessly even with limited storage and intermittent connectivity.

## Dashboard Prototype:

Below is a prototype for the sustainability data dashboard. The home page will provide general information about current emissions and waste reduction. On other parts of the dashboard, not pictured, the user will be able to find suggestions for sustainability initiatives specific to BU's current data. Alongside insights and recommendations, there will be a section of the dashboard that displays the many verifications being made from the data that is on the blockchain.

On the submitted GitHub page, there is a second prototype that is launched via Dash. Dash allows for data dashboard creation entirely using python. The file "DashPrototype.py" creates a simple graph using a sample csv file.





### Solidity Contract:

#### Code in DashContract.sol File

In this smart contract, the **postSensorData** function allows sensors to post their data to the blockchain, and the data is stored in a mapping with the sensor's address as the key. The **verifySensorData** function can be used to verify the integrity / authenticity of the posted data by providing the sensor vender's address, the index of the data in the mapping, the timestamp, the value, and a digital signature. The **recoverSigner** function is used to recover the signer's address from the provided digital signature. This way, users can verify that the displayed aggregate data results are based on the data from the blockchain and ensure its accuracy.

The photo on the left shows that the Solidity contract is both compiled and deployed using the Remix IDE.

### Code for fetching data, aggregating data, generating insights, and recommendations:

#### Code in dashboard\_backend.py file

This file currently includes pseudocode to the backend of the data dashboard. This includes aggregating the data from the blockchain, using the Solidity contract for data verification, as well as launching the data dashboard and providing the user with recommendations to possibly improve sustainability on campus.

## Bibliography

- Solidity verification
  - <https://solidity-by-example.org/signature/>
  - <https://xtremetom.medium.com/verifying-solidity-signatures-4898d003846b>
- Remix IDE
  - [https://remix-ide.readthedocs.io/en/latest/create\\_deploy.html](https://remix-ide.readthedocs.io/en/latest/create_deploy.html)
- Dashboard Implementation using Python libraries
  - <https://towardsdatascience.com/the-easiest-way-to-create-an-interactive-dashboar-d-in-python-77440f2511d1>
- Dash App (option for Dashboard Implementation)
  - <https://www.the-analytics.club/plotly-dashboards-in-python>
- Homeworks 5 and 7 (Circom and Solidity Contracts)