### **Detailed Individual Reflection – Behruz (Frontend Developer / UI/UX Designer)**

As the **Frontend Developer and UI/UX Designer**, my main responsibility was to create a static, interactive website where users could easily encrypt and decrypt text using different cryptographic algorithms. This meant building both the interface (what users see and interact with) and the logic that connects user inputs with algorithm outputs visually.

#### **🔨 What I Did:**

1. **Designed the Entire Website Interface**
   * I built the site from scratch using **HTML, CSS, and JavaScript**.
   * Implemented a **responsive layout** so it works on both desktop and mobile.
   * Added **text input areas**, **drag-and-drop upload**, and **output result boxes**.
2. **Integrated Algorithm Functions with the Interface**
   * I worked closely with Abbos (Algorithm Developer) to connect the algorithm logic to buttons like “Encrypt” and “Decrypt.”
   * For example, when a user enters text and clicks “Encrypt,” the frontend passes the data to the correct encryption function and displays all algorithm outputs below in a formatted style.
3. **Implemented File Upload (including Drag-and-Drop)**
   * I added a file input where users can upload .txt files.
   * Implemented drag-and-drop using dragenter, dragleave, dragover, and drop event listeners in JavaScript.
   * Initially, the drag-and-drop feature had a bug: sometimes the drop zone wouldn’t react visually when dragging a file over. I fixed this by adding proper event propagation handling and dynamic class toggling.
4. **Built Visual Enhancements for Education**
   * Added **step-by-step explanation boxes** for algorithms like Caesar and RSA.
   * For RSA, I created a **diagram** that shows how public/private key encryption works with arrows and stages.
   * Used **color coding** to highlight encryption steps (e.g., shift values in Caesar Cipher).
5. **Created a Clean, User-Friendly UI**
   * Used **flexbox and grid** layout to align all elements properly.
   * Styled buttons, text areas, and result sections using **CSS with modern aesthetics**.
   * Implemented a **Dark Mode** with toggle support and tested readability in both modes.
   * Added a **“Clear Output” button** based on feedback to help users reset the workspace.

#### **⚙️ How I Did It:**

* Used **Vanilla JavaScript** for DOM manipulation and event handling.
* Followed a **component-based layout**—separating the header, input section, algorithm display, and result area clearly.
* Collaborated in GitHub, testing changes locally before deployment.
* Regularly shared progress with the team via screenshots and demo links to gather feedback.

#### **🧩 What I Encountered:**

* **Drag-and-Drop Bug**: Initially, the file would be accepted but no visual indicator showed where to drop. This made the feature feel broken. I fixed it by preventing default behaviors and handling the dragenter/leave properly.
* **Responsive Design Issues**: On smaller screens, the layout broke (buttons were overlapping, inputs squished). I solved this by adding media queries and using flex-wrap for better mobile display.
* **Dark Mode Readability**: Some colors in dark mode (like blue and purple) were too dim on black backgrounds. I tested contrast ratios and chose colors that are readable and visually appealing.
* **Frontend-Backend Sync**: Making sure the encryption logic matched what the frontend expected. For example, RSA outputs were very long strings, so I formatted the results in a scrollable text box with a monospace font.

#### **🎓 What I Learned:**

* **Real-world UI/UX skills**: From building file upload to debugging layout issues, I gained hands-on experience in solving user interface problems that matter.
* **How to design for educational purposes**: I learned that it's not just about making things look good—it’s about making information **understandable** and **visually digestible**.
* **Team Collaboration**: I improved how I communicate with backend developers to integrate functionality smoothly.
* **How small details matter**: Something as simple as a “Clear Output” button or padding around elements makes a big difference in the overall user experience.

#### **✅ Example Features I Delivered:**

* A fully working UI for inputting text and files.
* Visual output for each algorithm in a clear, readable format.
* RSA encryption explanation with interactive diagram.
* Drag-and-drop file input that works across browsers.
* A polished Dark Mode for better accessibility.