**Name of the project:** Code Sanitizer-A tool that will clean all the unclean and redundant code of Figma Plugin.

**Group Members:**

Khushi Jobanputra,60004210147, C2

Manasvi Gupta,60004210235, C3

Kruti Shah, 60004210122 ,C2

Shashwat Shah,60004220126, C2

**Domain of Project:**

Machine leaning and Deep learning

**Problem Statement:**

Figma, a popular design tool among UI/UX designers, allows for the seamless creation of mobile applications and websites. However, the code generated by Figma plugins for languages like HTML, CSS, React, and Flutter often contains inefficiencies and redundancies. Examples of unclean code include an excessive number of divisions for single elements, lack of proper sequence, and the creation of unnecessary empty boxes. Such code can be problematic for developers, as it complicates the process of translating designs into functional and maintainable code, leading to increased development time and potential errors.

Our tool addresses this issue by automatically cleaning and optimizing the code generated by Figma plugins. It removes all useless and redundant code, ensuring a streamlined and efficient structure that adheres to the original design. By properly structuring the code according to the user's design, the tool not only enhances code readability but also significantly reduces the manual effort required by developers. This solution is particularly beneficial for teams aiming to accelerate their development process without compromising on code quality, ultimately facilitating a smoother transition from design to deployment.

**Research Gap:**

Despite the popularity and utility of Figma as a design tool among UI/UX designers, there is a significant research gap in the area of automatic code optimization and cleaning for the code generated by Figma plugins. While Figma excels at providing robust design capabilities and export options, the resulting code often contains inefficiencies and redundancies such as excessive divisions, empty boxes, and poorly organized structures. Current solutions are predominantly focused on code generation without offering robust tools for optimizing this code. This leaves developers with the time-consuming task of manually cleaning up and reorganizing the generated code, which can lead to increased development time and potential errors.

Moreover, there is limited research on maintaining design integrity during the code optimization process. Ensuring that cleaned and optimized code preserves the original design's visual and functional aspects is a complex challenge that has not been adequately addressed. Additionally, the variability in code structures across different languages like HTML, CSS, React, and Flutter adds another layer of complexity. A unified tool capable of handling these different languages while respecting their unique requirements is needed. Furthermore, existing research does not adequately explore user customization options to tailor the code cleaning process to specific needs, nor does it address effective integration strategies into existing development workflows. These gaps highlight the need for a comprehensive solution that automates code cleaning, preserves design integrity, supports multiple languages, and integrates seamlessly into development environments.

**Objectives:**

The objectives of the project are:

1. Code Optimization :Automatically clean and optimize the code generated by Figma plugins for HTML, CSS, React, and Flutter. Remove unnecessary divisions, empty boxes, and redundant elements to streamline the code structure.

2.Maintain Design Integrity :Ensure the cleaned code adheres to the original design created by the user in Figma. Preserve the visual and functional aspects of the design while optimizing the underlying code.

3. Enhance Readability: Improve the readability of the code by organizing it in a logical and structured manner. Follow best practices and conventions for each language to make the code more maintainable.

4. Reduce Development Time: Minimize the manual effort required by developers to clean and organize code .Accelerate the development process by providing ready-to-use, clean code that requires minimal adjustments.

5.Improve Code Quality: Eliminate common issues and inefficiencies in the generated code to enhance overall code quality. Ensure the code is efficient, performant, and free from unnecessary bloat.

6. User-Friendly Interface: Provide a simple and intuitive interface for users to input the generated code and receive the cleaned version. Offer customization options to allow users to tailor the cleaning process to their specific needs.

7. Support Multiple Languages: Ensure the tool can handle code generated for various languages, including HTML, CSS, React, and Flutter. Adapt the cleaning process to the unique requirements and conventions of each language.

8. Seamless Integration: Enable easy integration with existing workflows and development environments . Allow users to quickly incorporate the cleaned code into their projects with minimal disruption.

9. Continuous Improvement: Regularly update the tool to address new challenges and incorporate user feedback. Stay up-to-date with the latest best practices and standards in web and mobile development.

**Unique Contribution:**

The unique contribution of this project lies in its ability to automatically clean and optimize code generated by Figma plugins, specifically targeting HTML, CSS, React, and Flutter. Unlike existing solutions that primarily focus on code generation, our tool will go a step further by removing unnecessary divisions, empty boxes, and redundant elements, resulting in more streamlined and efficient code. This automated approach significantly reduces the manual effort required by developers to clean up and organize the code, thereby accelerating the development process and enhancing code quality. By ensuring that the cleaned code adheres to best practices and conventions for each language, our tool will improve readability and maintainability, making it easier for developers to work with the code.

Additionally, our tool will maintain the integrity of the original design created in Figma, balancing code optimization with the preservation of visual and functional aspects. This is particularly important for ensuring that the final implementation accurately reflects the designer's intent. The tool's ability to support multiple languages, including HTML, CSS, React, and Flutter, and adapt the optimization process to the unique requirements of each, sets it apart from other solutions. Furthermore, by offering customization options for users to tailor the cleaning process to their specific needs, our tool provides a flexible and user-friendly solution. This comprehensive approach not only addresses the inefficiencies in the code generation process but also integrates seamlessly into existing development workflows, ultimately enhancing productivity and code quality in web and mobile development projects.

**In – house/out – house project:**

It is an In-house project.