**Milestone 2 - Report**

**Submitted by:  
Shai Kikozashvily – 206202384  
Oren Elbazis - 204864532**

**Project summary**:

Our project aims to simplify 3D printing for users by addressing the complexity of traditional 3D modeling software. Recognizing the challenge many faces in learning such software, the project proposes a chatbot-based solution, called "From Text to 3D using LLM." This system allows users to input plain text descriptions of desired 3D objects, which are then interpreted, visualized, and modified interactively before generating printer-ready code. Unlike existing platforms like "Addithive", which solely generate code from text descriptions, this project offers an innovative approach by providing users with a virtual 3D model that they can modify before printing. Leveraging a multi-agent system with a finetuned GPT-4 model, the project ensures efficient translation of user prompts into 3D printing instructions (G-code).

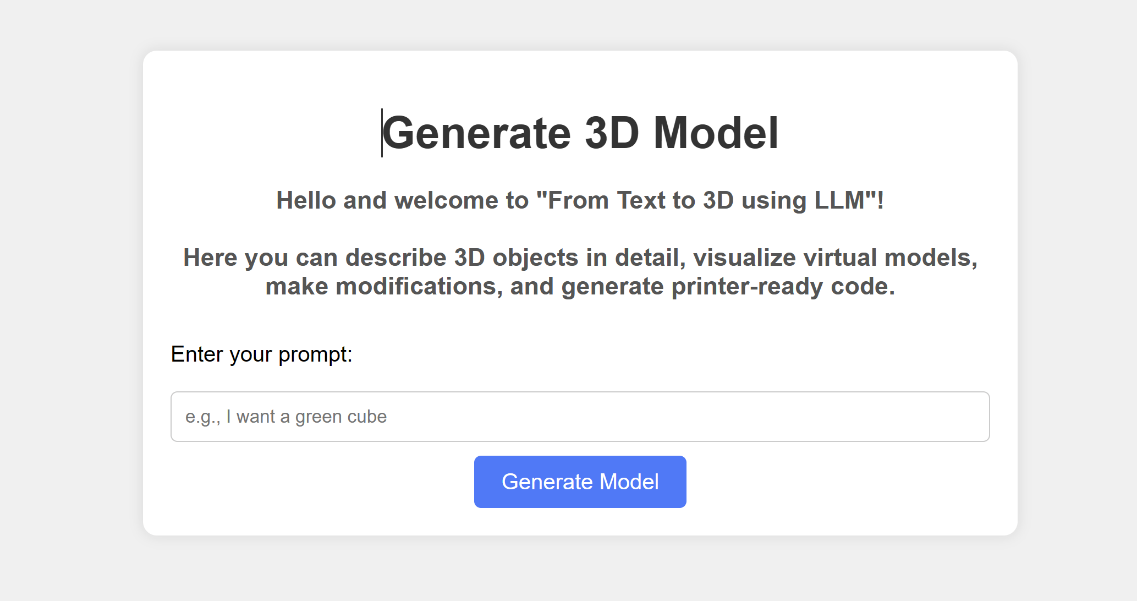
**Design principles**:

Our design is guided by several principles for human-AI interaction:

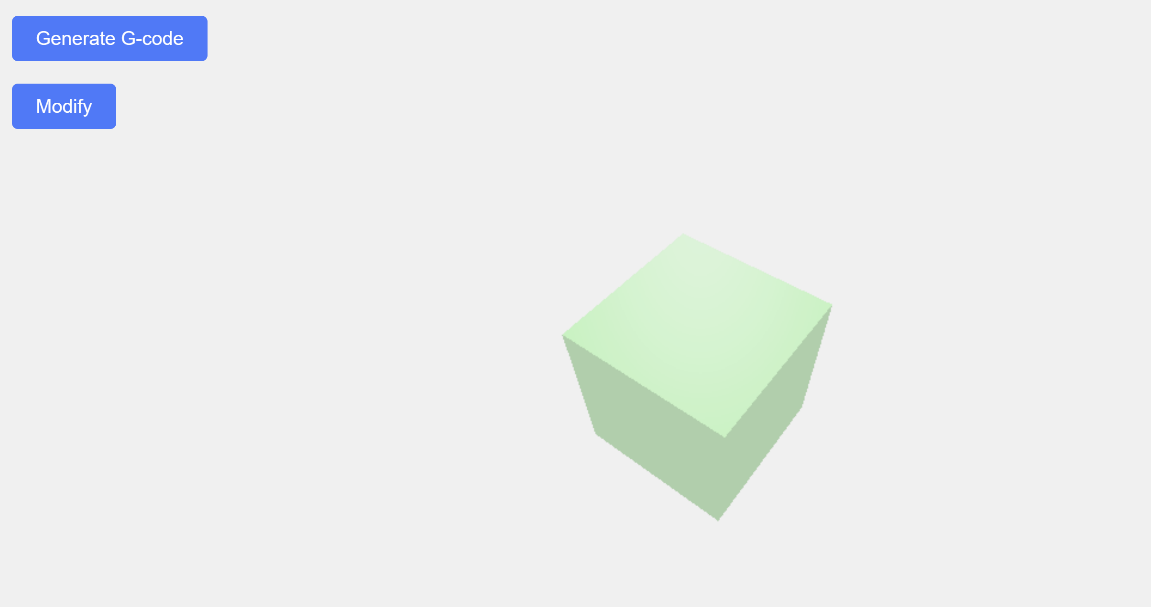
1. **Make clear what the system can do:** Our project clearly outlines its capabilities, informing users that it interprets detailed descriptions of 3D objects, presents virtual models, allows modifications, and generates printer-ready code.
2. **Time services based on context:** Our system adapts its services based on user interactions and context. For instance, it provides immediate feedback during description interpretation and model visualization, ensuring timely responses throughout the user journey.
3. **Show contextually relevant information:** Contextually relevant information, such as user input, model modifications, and 3D model visualization, is displayed in a clear and accessible manner, enhancing user understanding and engagement.

**Interface instruction**:

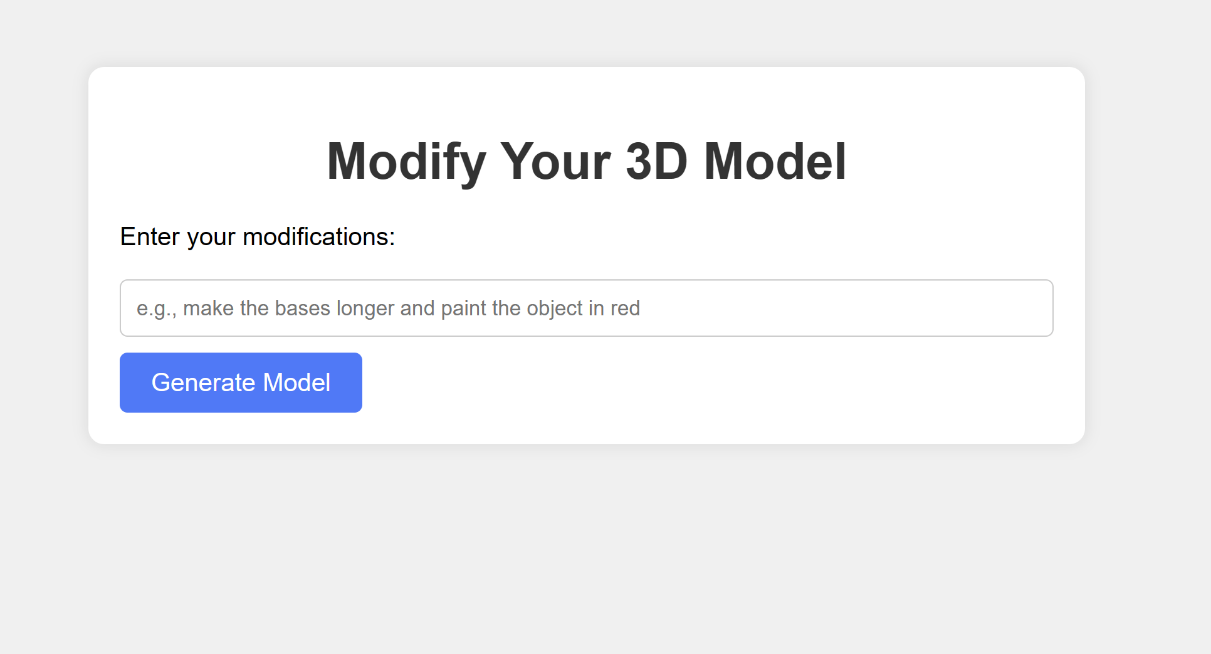
* Interpretation: The system understands detailed descriptions provided bythe user regarding the 3D object they wish to print.



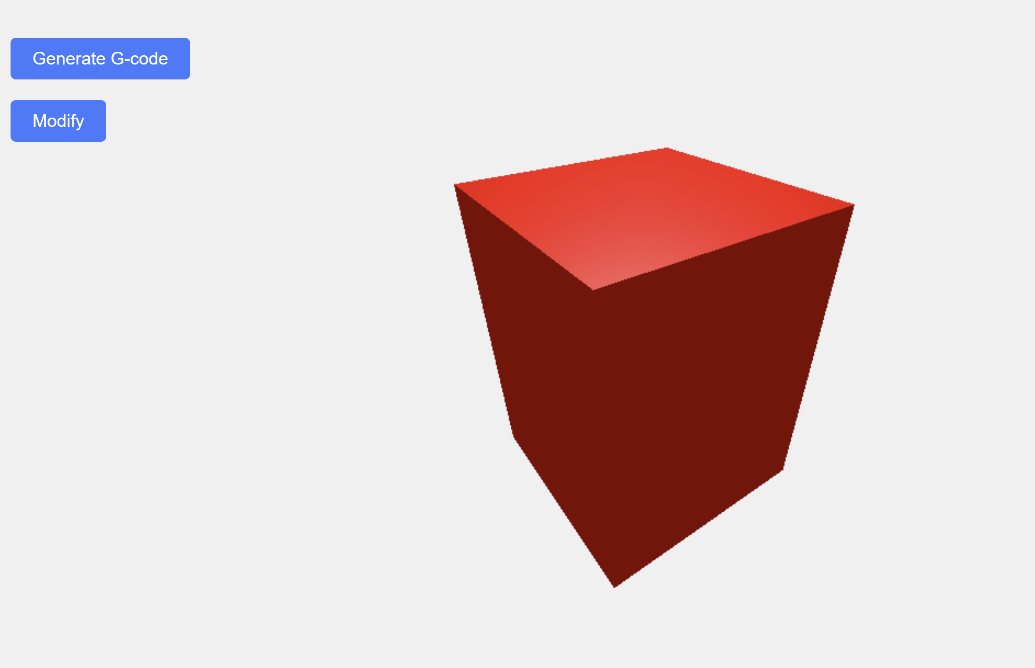
• Visualization: After interpreting the description, the system presents a virtual 3D model of the object. This allows the user to see a representation of their idea before it is printed.



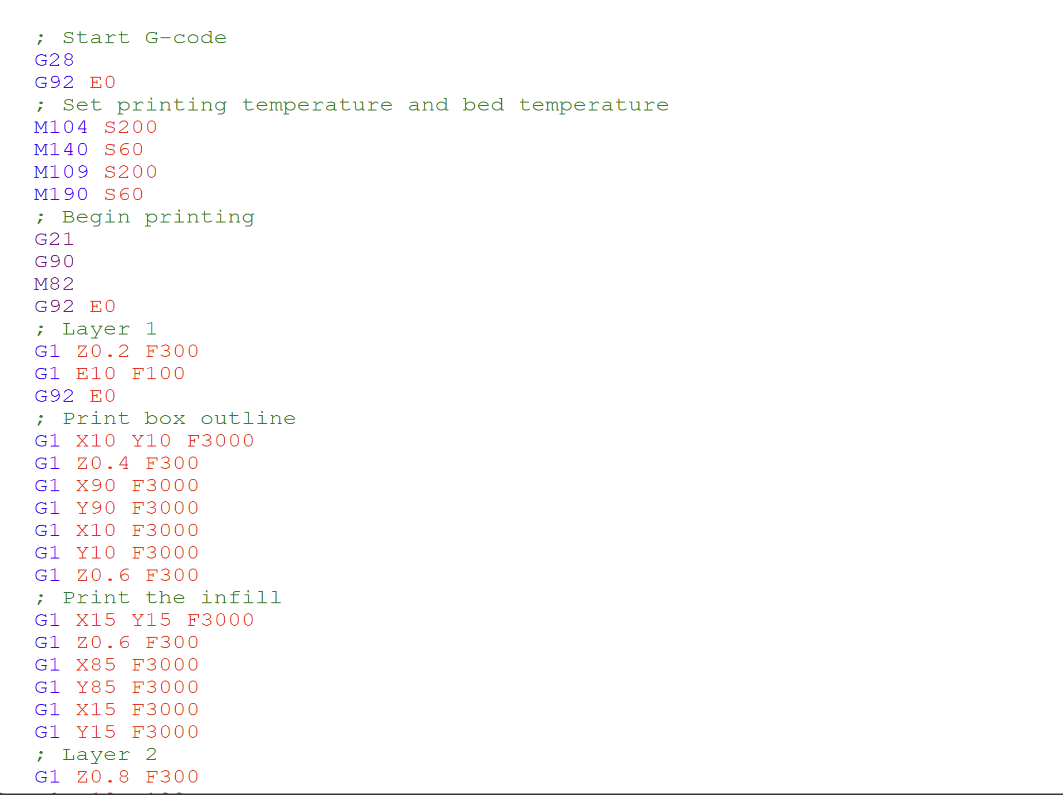
• Modification: Users will have the ability to make adjustments to the 3D model. This interactive feature ensures that the final product matches their expectations and requirements



• Visualization: After modifying the description, the system presents a modified virtual 3D model of the object.



• Code Generation: Once the user is satisfied with the 3D model, the system generates the necessary code that can be directly used by a 3D printer to create the physical object.



**Link to a video demonstration:**

<https://1drv.ms/v/s!AsSgwRzNAswK0wcm0wFUe4byvNWl>

**Algorithmic approach**:

1. **Multi-Agent System - leveraging a finetuned GPT-4 model**: Thisfinetuning specifically focuses on 3D printer code generation, enabling themodel to effectively parse user prompts and construct viable 3D printinginstructions. The LLM agents will work in a collaborative pipeline, where eachagent specializes in different aspects of the translation process, from initialtext interpretation to the final 3D printer-ready code output.
2. **Communication between Python and Rhino:** We explore various approachesto establish communication between Python and Rhino, a 3D modeling software. One approach involves utilizing the RhinoPython library for direct integration, enabling seamless data exchange and allowing Python scripts to control Rhino's functionalities and manipulate 3D models according to user requirements. Additionally, we consider alternative methods such as using intermediary file formats or web APIs to facilitate communication between the two platforms. Each approach is evaluated based on factors like efficiency and ease of implementation.
3. **Web Interface Development**: We use Flask, a lightweight web framework for Python, to develop the web interface. Flask enables us to create interactive user interfaces where users can input text descriptions, visualize 3D models, and interact with the system seamlessly.
4. **3D Visualization**: For 3D visualization, we utilize Three.js, a JavaScript library that enables the creation and display of 3D content in web browsers. Three.js facilitates the rendering of virtual 3D models based on user descriptions, allowing users to visualize and interact with their designs in real-time.