

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Інформаційна технологія слайсингу моделей для FDM 3D
принтера»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проектування»

Виконавець роботи: студент групи ІТ.м-01 Новохатько Ігор Олександрович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

_____ «__» грудня 2021 р.

Науковий керівник

_____ к.т.н., доцент Парфененко Ю.В.
(підпис)

Голова комісії

_____ Шифрін Д.М.
(підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«__» _____ 2021 р.

1 ЗАВДАННЯ
2 НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА СТУДЕНТУ

Новохатько Ігор Олександрович

1 Тема роботи Інформаційна технологія слайсингу моделей для FDM 3D принтера

керівник роботи к.т.н., доцент Парфененко Юлія Вікторівна,

затверджені наказом по університету від «29» 10 2021 р. № 0787-IV

2 Строк подання студентом роботи «10» грудня 2021 р.

3 Вхідні дані до роботи Методика триангуляції 3D моделі, методика згладжування 3D моделі, методика слайсингу 3D моделі, методика генерації підтримок

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз предметної області, 2) Проектування інформаційної технології слайсингу моделей для FDM 3D принтера, 3) Розробка інформаційної технології слайсингу моделей для FDM 3D принтера

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) аналіз предметної області, мета і задачі дипломного проекту, порівняння аналогів, функціональні вимоги до інформаційної системи, моделювання роботи, декомпозиція першого рівня, декомпозиція під процесів, діаграма варіантів використання, архітектура додатку, засоби реалізації, демонстрація додатку, висновки, оприлюднення роботи

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Оформлення планування робіт	До 20.10.2021	
2.	Оформлення технічного завдання	До 06.11.2021	
3.	Проведення аналізу предметної області	До 10.11.2021	
4.	Проведення структурно-функціонально моделювання процесів	До 12.11.2021	
5.	Розробка додатку	До 24.11.2021	
6.	Тестування додатку	До 30.11.2021	
7.	Здача пояснювальної записки та файлів розробленого проекту	До 07.12.2021	

Студент

(підпис)

Новохатько І.О.

Керівник роботи

(підпис)

Парфененко Ю.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Інформаційна технологія слайсингу моделей для FDM 3D принтера».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 30 найменувань, 2 додатків. Загальний обсяг роботи – 58 сторінок, у тому числі 34 сторінки основного тексту, 2 сторінки списку використаних джерел, 22 сторінки додатків.

Кваліфікаційну роботу магістра присвячено розробці інформаційної технології слайсингу моделей для FDM 3D принтера.

У роботі проведено огляд останніх досліджень та публікацій, які стосуються 3D друку, аналіз аналогів, моделювання та проектування додатку.

У роботі виконано розробку web-додатку на основі інформаційної системи. Також було розроблено модулі завантаження моделі, відтворення завантаженої моделі, генерації системи підтримок, збереження отриманих результатів, слайсингу 3D моделі, особистого кабінету користувача, адміністратора системи.

Результатом проведеної роботи є розроблена інформаційна технологія слайсингу моделей для FDM 3D принтера.

Розроблена інформаційна технологія впроваджена у web-додатку який призначений для слайсингу моделей для FDM 3D принтера, для користування яким потрібно мати доступ до мережі та смартфон або ноутбук, тобто фактично з будь-якої точки світу можна буде використовувати web-додаток.

Ключові слова: web-додаток, слайсинг, триангуляція, WebGL.

ЗМІСТ

ВСТУП	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	5
1.1 Огляд останніх досліджень і публікацій	5
1.2 Аналіз проблеми забезпечення якості друкованих виробів на етапі слайсингу 3D моделей.....	8
2 ПОСТАНОВКА ЗАДАЧІ.....	11
2.1 Мета й задачі дослідження.....	11
2.2 Алгоритм (модель) виявлення вразливих місць/додавання системи підтримок	Ошибка! Закладка не определена.
2.3 Моделювання роботи інформаційної системи в IDEF0.....	15
2.4 Моделювання варіантів використання інформаційної технології слайсингу моделей для FDM 3D принтера.....	19
2.5 Проектування моделі бази даних.....	21
3 РОЗРОБКА WEB-ДОДАТКУ ДЛЯ НАЛАШТУВАННЯ МОДЕЛІ ПЕРЕД ДРУКОМ НА 3D-ПРИНТЕРІ.....	22
3.1 Архітектура web-додатку.....	22
3.2 Програмна реалізація	23
3.3 Використання програмного додатку	27
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	36
ДОДАТОК А. ПЛАНУВАННЯ РОБІТ	39
ДОДАТОК Б	47

ВСТУП

3D-друк – процес створення цілих тривимірних об'єктів практично будь-якої геометричної форми з урахуванням цифрової моделі. 3D-друк базується на концепції виготовленні об'єкта пошарово. Практично, 3D-друк дає собою досконалу протилежність класичних способів механічного виготовлення та обробки, як фрезерування або різання, де створення виробу відбувається за рахунок видалення зайвого матеріалу.

3D-принтерами називають верстати з програмним керуванням, що виконують побудову деталі адитивним способом.

Слайсер – комп'ютерна програма, що пошарово перетворює 3D-модель в машинний код (G-code), що дозволяє виготовити деталь, використовуючи спеціалізований матеріал.

Залежно від технології пошарового або поверхневого формування деталі результатом роботи можуть бути файли, що містять способи формування шарів – векторні лінії, растрові плашки, шляхи переміщення, нормалі до поверхні. Необхідність розроблення інформаційної технології полягає в тому, що популярність 3D друку дуже швидко поширюється і знайти додатки, які допоможуть спростити процес підготовки моделі складно, до того ж чим більший вибір між додатками у користувачів тим, більше покращень буде реалізовано розробниками.

Об'єкт дослідження – інформаційне забезпечення створення та підготовки тривимірних моделей для 3D друку.

Предмет дослідження – процес слайсингу тривимірних моделей для друку на 3D принтері.

Метою кваліфікаційної роботи є розробка інформаційної технології слайсингу моделей для FDM 3D принтера, а також сканування моделі на наявність різного виду помилок їх візуальне відображення та створення системи підтримок.

Задачами являються:

- аналіз предметної області та вибір методів дослідження;
- вибір методів відображення помилок моделі;
- створення та вибір патернів для системи підтримок;
- створення інформаційної технології слайсингу моделей для FDM 3D принтера;
- розробка програмного додатку для тестування роботи інформаційної технології.

Актуальність інформаційної технології слайсингу моделей для FDM 3D принтера полягає в необхідності виконання деяких функції обробки 3D моделі, наявних в платних спеціалізованих програмах, які не є широкодоступними.

Практичне значення полягатиме у використанні інформаційної технології при реалізації програмного додатку, що буде використовуватися в якості редактора та слайсера для 3D моделей, результати роботи якої можуть бути використані для друку на 3D принтері.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Огляд останніх досліджень і публікацій

Технології 3D-друку використовуються для прототипування та розподіленого виробництва в архітектурі, будівництві, промисловому дизайні, автомобільній, аерокосмічній, військово-промисловій, інженерній та медичній галузях, біоінженерії (для створення штучних тканин), виробництві модного одягу та взуття, ювелірних виробів, в освіті, географічних інформаційних системах, харчової промисловості та багатьох інших сферах [1].

3D-моделі створюються методом ручного комп'ютерного графічного дизайну або 3D-сканування [2].

3D-сканування – це автоматичний збір та аналіз даних реального об'єкта, а саме форми, кольору та інших характеристик, з подальшим перетворенням на цифрову тривимірну модель.

Під час друку принтер зчитує 3D-друкований файл (зазвичай у форматі STL), що містить дані тривимірної моделі, і наносить послідовні шари рідкого, порошкоподібного, паперового або листового матеріалу, вибудовуючи тривимірну модель із серії поперечних перерізів [3]. Ці шари, що відповідають віртуальним поперечним перерізам CAD-моделі, з'єднуються або сплавляються разом для створення об'єкта заданої форми. Основною перевагою цього методу є можливість створення геометричних форм практично необмеженої складності. Для підготовки 3D моделей до друку необхідно спеціальне програмне забезпечення, слайсер, направлене на конкретний тип принтеру або на групу принтерів з відкритим вихідним кодом [4].

Слайсери поділяються на два основні види: універсальні та спеціалізовані (корпоративні). Зазвичай спеціалізовані «заточені» під одну технологію, торгову марку або модельну лінійку принтерів. Універсальні мають велику варіативність у налаштуваннях та розраховані на широкий спектр сумісних пристроїв [5].

Деякі виробники вважають за краще використовувати широко відомі універсальні слайсери, доповнивши їх спеціалізованими профілями свого обладнання. Інші інвестують кошти в розробку власного програмного забезпечення, як правило, жорстко «прив'язане» до їх обладнання. Хоча бувають винятки, наприклад, розробники та виробники принтерів Raise3D свій IdeaMaker, підготовлений для роботи з рідною «апаратурою», дозволяють налаштувати на будь-який сумісний принтер [6].

Для правильного підключення з програмою необхідно налаштувати принтер, прописавши деяку кількість специфічних параметрів. Базові принципи роботи та результати слайсингу у всіх програм нарізки багато в чому ідентичні. Можна виявити такі відмінності як слайсинг, інтерфейси, кількість доступних налаштувань, ресурсоємність програми, швидкість її роботи і додаткові сервіси, які, часом, є маркетинговими прийомами.

Інша сегментація слайсерів описується їх доступністю для користувача. Вони бувають платними та безкоштовними. У FDM технології універсальною платною програмою залишилася лише Simplify3D [7].

Розробники намагаються створити інтуїтивно зрозумілі взаємозв'язки між основними блоками налаштувань: принтер, модель, матеріал, профіль друку, екструдери, додаткові опції, скрипти та макроси [8].

Структурно всі слайсери, крім головного обчислювального ядра програми, що відповідає за математичні розрахунки геометричних форм та конвертацію їх у мову машинного керування g-code, мають стандартні блоки налаштувань [9].

На рис. 1.1 можна побачити весь процес 3D друку від моменту вибору методу отримання необхідної моделі до моменту постобробки готового виробу.

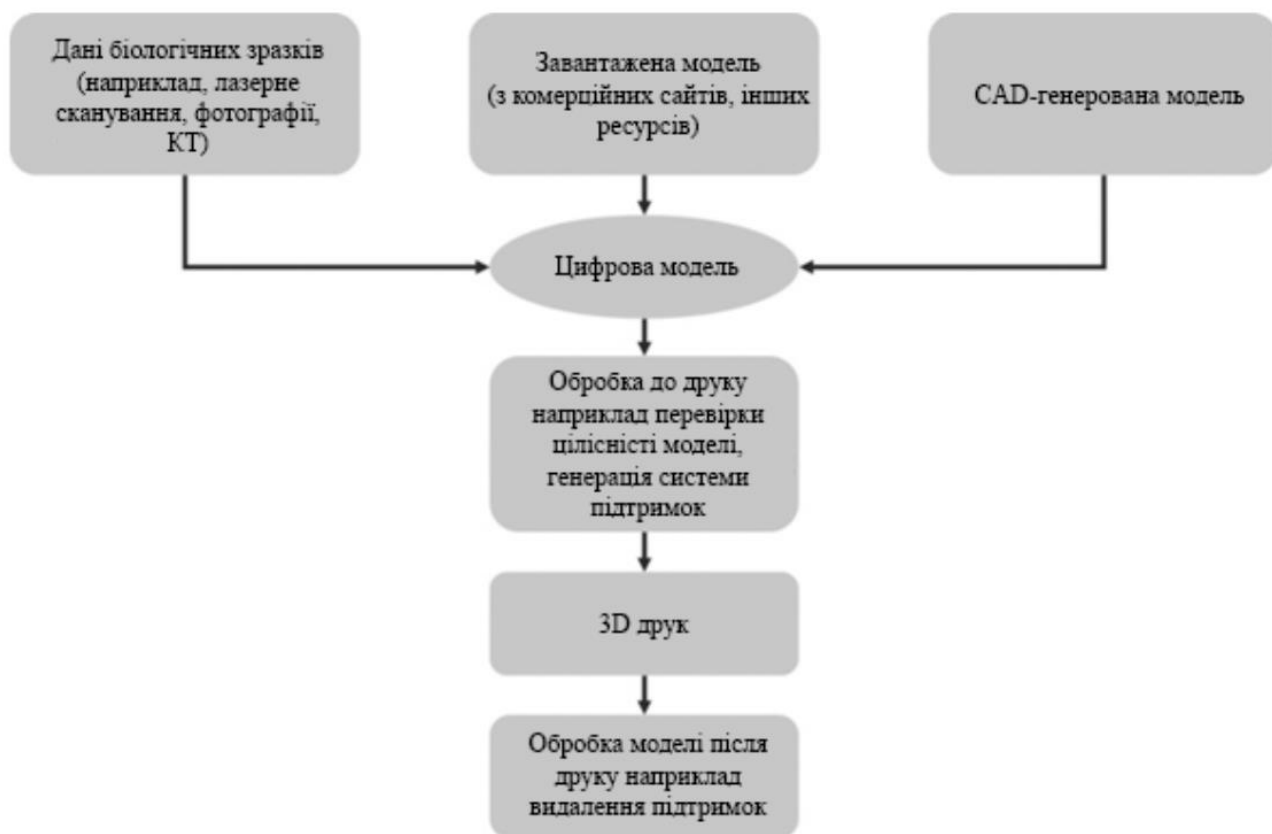


Рисунок 1.1 – Структурна схема 3D друку

Першим етапом структурної схеми є способи отримання 3D моделі, в блоках вказано, що модель може бути отримана в результаті 3D сканування, завантажена з відкритих ресурсів або згенерована програмою.

Наступним етапом є завантаження моделі до інформаційної технології.

Третій етап є важливим етапом всієї схеми, саме в ньому відбувається аналіз моделі на предмет помилок та варіантів їх усунення, також включаючи процес створення системи підтримок. Сама система підтримок має велику ефективність, так як дозволяє друкувати певні форми (рис. 1.4), але в той самий час вона несе з собою проблему більшої витрати матеріалу заради якості. Взагалі третій етап включає в себе підготовку моделі до друку. Від параметрів слайсингу та системи підтримок залежить час друку та якість надрукованої моделі.

Четвертим етапом є процес друку, коли підготовлена модель відправляється на контролер принтеру та друкується.

Останній етап являє собою обробкою надрукованої моделі, зняття її з робочої поверхні та видалення системи підтримок також подальша обробка.

Інформаційна технологія розроблятиметься для 3 блоку структурної схеми, представленої на рис. 1.1.

1.2. Аналіз проблеми забезпечення якості друківаних виробів на етапі слайсингу 3D моделей

FDM 3D-принтери працюють шляхом нанесення шару на шар термопласту для створення 3D-об'єкта. У цьому методі кожен новий шар повинен підтримуватися шаром під ним. Якщо модель має звис, який не підтримується чимось нижче, потрібно додати додаткові структури підтримки 3D-друку, щоб забезпечити успішний друк [10].

Підтримуючі конструкції вважаються необхідним у 3D-друці. З одного боку, вони необхідні для моделей з звисами або містками. З іншого боку, вони збільшують матеріальні витрати, додають більше роботи після обробки та можуть пошкодити поверхню моделі. Тому правильна підтримка конструкцій для 3D-друку є дуже важливим аспектом складних моделей 3D-друку [11].

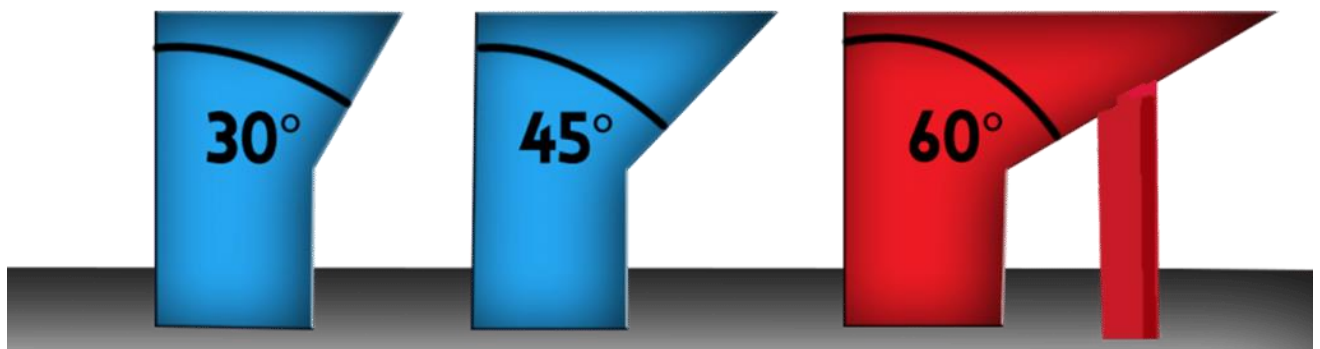


Рисунок 1.2 – Допустимий кут друку (зліва) та недопустимий кут (справа)

Тому для друку моделей, які мають недопустимий кут, використовується система підтримок (рис. 1.3).

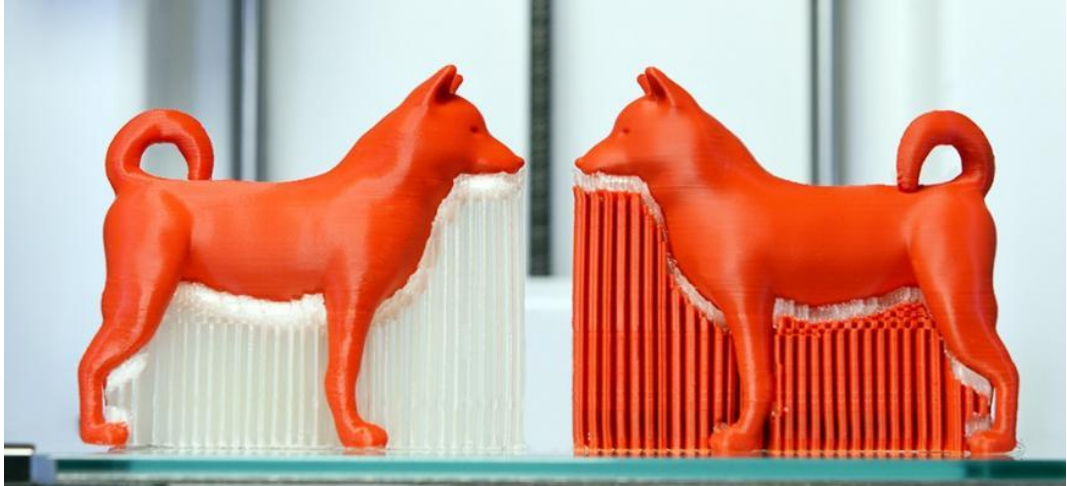


Рисунок 1.3 – Надрукована модель з системою підтримок

З іншого боку, звиси літери Т мають кут 90 градусів. Таким чином, ви повинні використовувати допоміжні структури для 3D-друку [12], щоб надрукувати літеру Т, інакше результат буде таким, як показано на рис. 1.4.



Рисунок 1.4 – Надрукована модель з системою підтримок та без

Якщо потрібна система підтримок, можна встановити прапорець «Enable Support» в панелі параметрів та побачити результат. Тим не менш, існує багато шаблонів системи підтримок 3D-друку [13].

Формування системи підтримок використовує патерни для створення певних типів підтримки, вибір патерну залежить від типу моделі та інших факторів [14]. Автоматичне створення системи підтримок відбувається з врахуванням кута відносно друкуючої поверхні [15].

Дана кваліфікаційна робота магістра спрямована на спрощення процедури підготовки моделі до друку, покращення якості друку моделей шляхом створення додаткових структур.

2 ПОСТАНОВКА ЗАДАЧІ

2.1. Мета й задачі дослідження

У ході виконання кваліфікаційної роботи магістра буде розроблено інформаційну технологію слайсингу моделей для FDM 3D принтера, а також аналізу моделі на наявність браку.

Інформаційна технологія повинна виконувати ряд задач:

- створення системи підтримок;
- налаштування системи підтримок.

Для реалізації поставленого завдання були виділені задачі для дослідження.

- розробка модулю створення системи підтримок;
- розробка модулю відображення системи підтримок;
- відладка та тестування дозволить виявити помилки та недоліки.

Перераховані вище задачі можна розглянути більш детально.

Створення карти вразливих місць буде працювати автоматично. Відразу, як користувач зайдє на ресурс та завантажить власну модель буде проведено аналіз, вразливі місця потребують наявності системи підтримок для нормального друку.

Система підтримок в свою чергу буде застосована лише по бажанню користувача, для її активації в бічному меню буде необхідний флаг активації. Якщо користувачу необхідна система підтримок, то після активації буде автоматичне створення системи підтримок. Автоматичне створення системи підтримок буде генерувати підтримки в місцях, які згідно аналізу не зможуть надрукуватися правильно [16].

На основі створеної інформаційної технології буде впроваджено у web-додаток для розширення функціональних можливостей систему підтримок для покращення 3D друку.

Для реалізації інформаційної технології буде використано JavaScript бібліотеку Three.js. Вона містить вибрані класи для розробок для створення і відтворення 3D графіки в WebGL. Бібліотека Three.js полегшує роботу з WebGL. При використанні Three.js немає необхідності писати шейдери. Сама бібліотека написана на javascript і призначена для використання в середовищі javascript. Вона буде запускатися на стороні клієнта – у веб-браузері на якомусь пристрої. Three.js постачається з власною математичною бібліотекою з конкретними класами для 3D математики.

Також буде використано earcut.js – ця бібліотека реалізує модифікований алгоритм нарізки моделі, оптимізований за допомогою хешування кривої z-порядку та розширена для обробки отворів, скручених багатокутників, вироджень і самоперетину таким чином, що не гарантує правильність триангуляції.

2.2. Алгоритм слайсингу моделей

Алгоритм зрізу – це метод перетворення кожної трикутної грані на кожен відповідний сегмент лінії. Ці сегменти ліній можуть бути з'єднані в контурні лінії за допомогою алгоритму генерації контурів. Грані у форматі STL є довільними та мають різну орієнтацію [18].

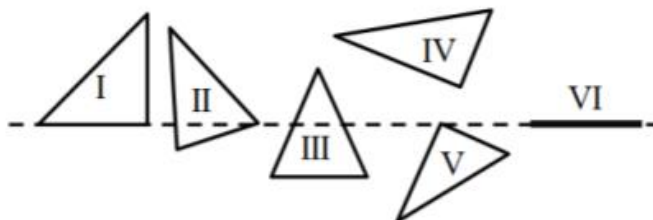


Рисунок 2.1 – Грані трикутників

На рис. 2.1 показані, а в таблиці 2.1 описані всі можливі взаємодії між площиною зрізу і гранями. Кожна взаємодія має бути оброблена належним чином, тому що для відрізка лінії потрібно лише 2 окремі точки.

Алгоритм перевіряє, чи є ребро паралельним або перетинається з площиною, щоб ізолювати випадок коли немає перетину, і усуває ребро яке збігається у випадку коли лінія проходить через один бік грані. Випадок, коли лінія, що ділить грань навпіл через одну вершину, обробляється шляхом усунення однієї повторюваної точки перетину. Перетин вершин може бути ідентифікований, коли існують дві точки перетину, які схожі.

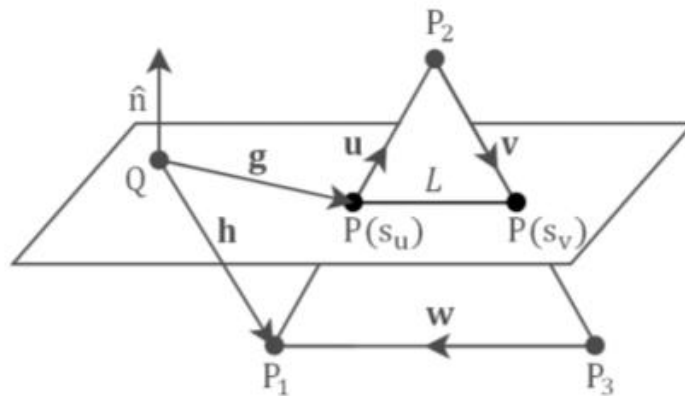


Рисунок 2.2 – Діаграма перетину граней та площин

Алгоритм зрізу в основному базується на рівнянні перетину прямої та площини. Параметричне рівняння прямої може бути записане у вигляді:

$$P(s) = P_0 + s(P_f - P_0) \text{ та } Q = (x, y, z_{\text{slice}})$$

де X та Y можуть бути будь-якою точкою на площині (зазвичай встановлюється на початку координат), Z_{slice} є висотою зрізу, а одиничний вектор $n = \{0, 0, 1\}$ є нормаллю площини (для випадку зрізу щодо осі Z).

Зміна значення Z_{slice} вплине на висоту зрізу та приведе до іншого сегменту лінії. Алгоритм спочатку перевіряє, чи існує перетин між лініями та площиною, використовуючи критерій точок, $n * u = 0$, $n * v = 0$, $n * w = 0$. Ці критерії також будуть обробляти як випадок I, так і випадок VI, усуваючи паралельну лінію в фасеті.

На рис. 2.2, u, v, i, w – вектори напрямку для кожної сторони грані за годинниковою стрілкою ($P1, P2, P3, P1$). Добуток точок стане рівним нулю, якщо немає перетину, що також означає, що лінія паралельна площині і не перетинатиметься з площиною. Лінія повинна перетинатися з площиною, такою як лінія між точкою $P1$ і $P2$ де $P_o = P1, P_f = P2$ і $s = su, n * u \neq 0$. У точці перетину вектор напрямки від $P(s)$ до Q та $g = h + su$, де вектор $h = P_o - Q$. Оскільки вектор $g = P(s) - Q$ лежить на площині, можна використовувати попередній критерій, тому що відомо, що $n * g = n * (h + su) = 0$. Вирішення цієї проблеми призведе до:

$$s = \frac{-n * h}{n * u} = \frac{n * (Q - P_o)}{n * (P_f - P_o)}$$

Використовуючи попередні рівняння та значення, отримане з іншого, рівняння набуває вигляду:

$$P(Su) = P1 + Su(P2 - P1)$$

Однак, значення має бути перевірене на відповідність діапазону $0 < s < 1$, щоб переконатися, що точка перетину існує тільки в межах лінії. Також метод застосовується до інших рядків. На рис. 2.2 $P(Su)$ і $P(Sv)$ генерують сегмент L лінії, який використовуватиметься в алгоритмі генерації контурів.

Таблиця 2.1 – Випадки взаємодії

Випадок	Взаємодії грані	Можлива точка взаємодії
I	Лінія проходить через один бік грані	4
II	Лінія, що ділить грань навпіл через одну вершину	3
III	Лінія, що ділить грань навпіл через дві сторони	2
IV	Відсутній перетин	0
V	Перетин вершин	2
VI	Паралельне перетин	6

2.3. Моделювання роботи інформаційної системи в IDEF0

Процес проектування інформаційної технології необхідно розпочинати з розробки контекстної діаграми IDEF0 [16]. Дане рішення прийнято через те, що потрібно розуміти чіткі та зрозумілі твердження, що описують систему її взаємодію з навколишнім середовищем. До контекстної діаграми входять:

- Вхід: дані, що використовуються системою для отримання результату(вихідних даних).
- Вихід: дані, які отримуються в результаті роботи системи.
- Управління: інформація, управляюча діями системи.
- Механізми: ресурси, що виконують роботу в системі.

Проаналізувавши головні елементи контекстної діаграми, для налаштування моделі перед друком на 3D-принтері, на основі цього сформовано наступний перелік:

- Вхід: файл 3D-моделі в форматі stl.
- Вихід: результат слайсованої моделі.
- Управління: методика триангуляції моделі, вимоги до слайсованої моделі, вимоги до збереження даних, алгоритм до генерації системи підтримок.
- Механізми: користувач, web-додаток, WebGL.

На основі цих даних була розроблена контекстна діаграма, що представлена на рис. 2.1.

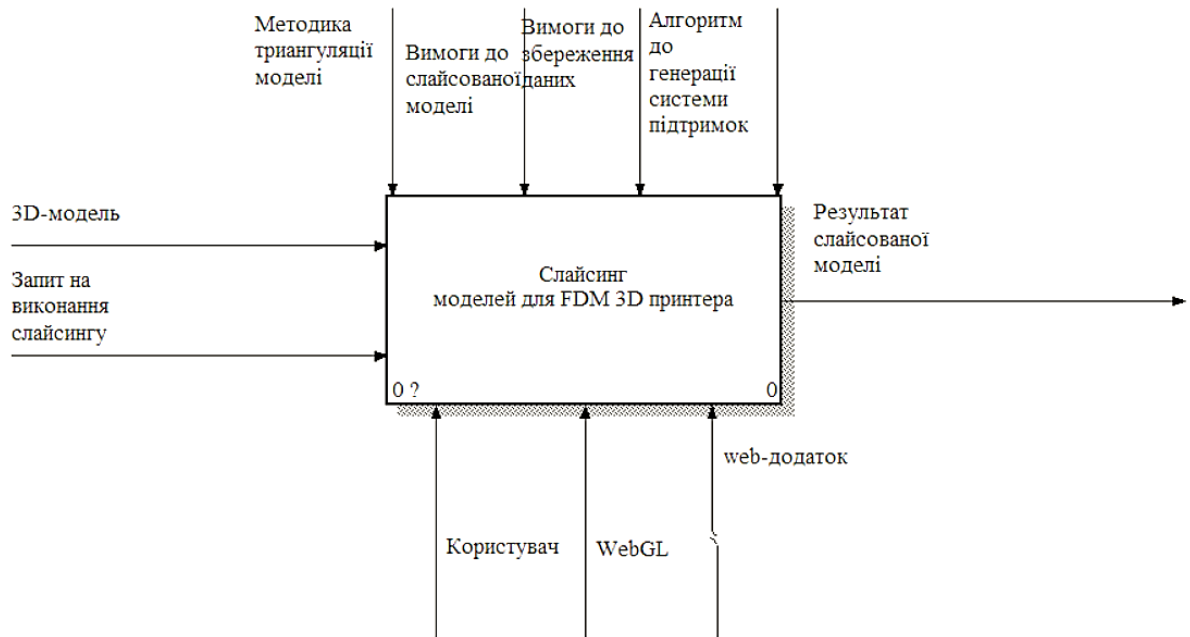


Рисунок 2.1 – Контекстна діаграма

Таке представлення показує лише загальні можливості. Для того, щоб побачити більш деталізовану модель, необхідно зробити декомпозицію. Дана діаграма показує функціональну модель інформаційної технології слайсингу моделей для FDM 3D принтера.

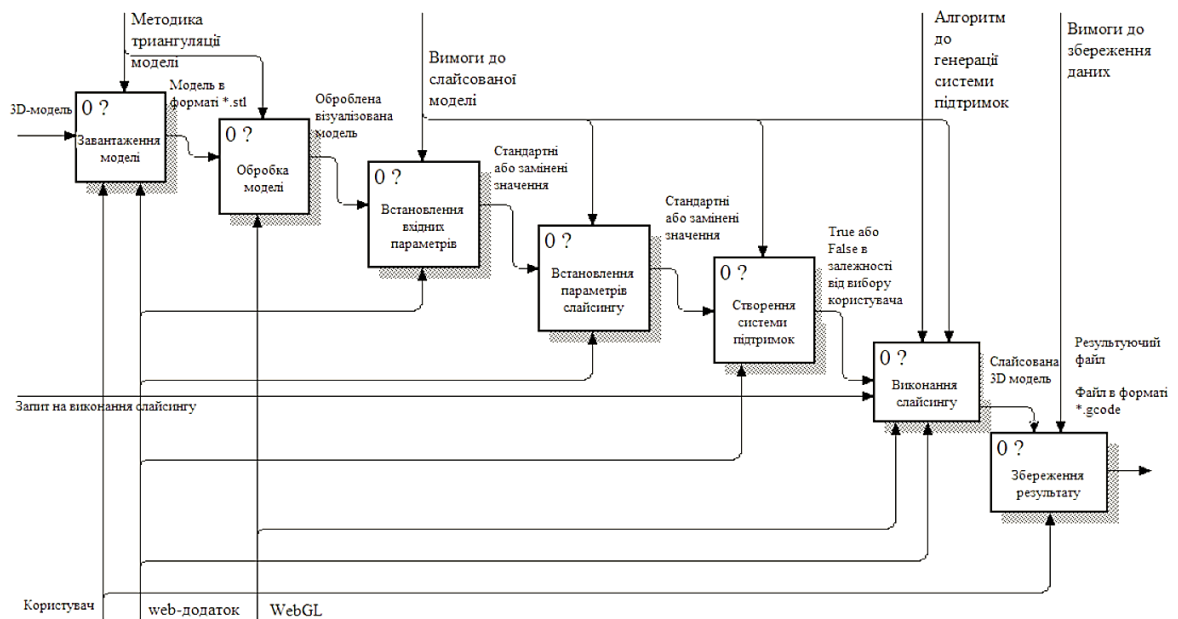


Рисунок 2.2 – Декомпозиція контекстної діаграми – функціональна модель інформаційної технології

Детальний опис: спочатку користувач потрапляє на web-ресурс, завантажує власну 3D-модель, після чого відбувається обробка моделі та її візуальне відображення. Для того, щоб отримати необхідний результат, користувач вводить параметри для правильного слайсингу моделі. Якщо модель потребує систему підтримок, користувачу потрібно встановити необхідний параметр. Після натискання кнопки слайсингу модель обробляється згідно введеним користувачем параметрам, в результаті користувач одержує відслайсовану модель в форматі *.gcode, яку зможе завантажити та використати для друку на 3D принтері типу FDM.

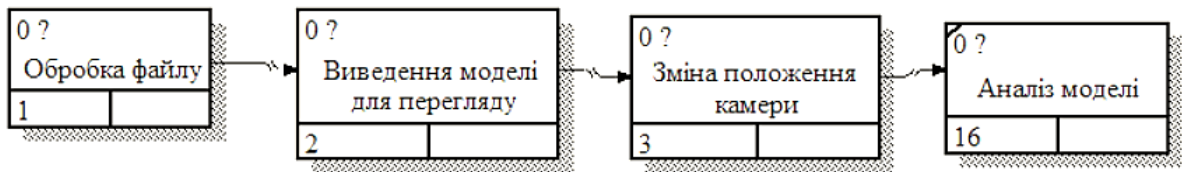


Рисунок 2.3 – Декомпозиція блоку 2 “Обробки моделі ” в нотації IDEF3

Спочатку модель потрапляє на обробку після чого відображається, користувач змінює положення камери оглядаючи модель, аналіз моделі відбувається згодом (рис. 2.3).

Проаналізована та оброблена модель очікує коли користувач задасть параметри друку та системи підтримок після того як користувач відправить модель на обробку буде отримано результат (рис. 2.4).

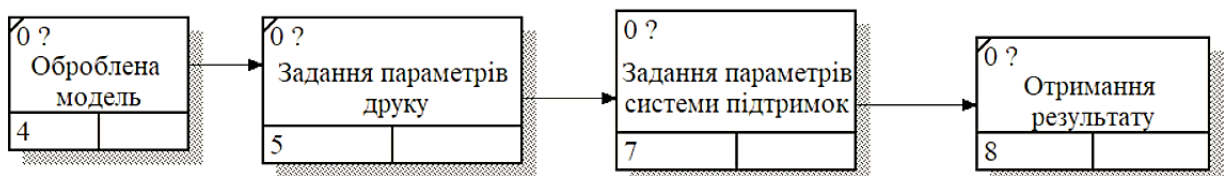


Рисунок 2.4 – Декомпозиція підпроцесу “Створення системи підтримок” в нотації IDEF3

Параметри слайсингу передаються алгоритму слайсингу після чого відбувається парсинг моделі в формат gcode та завантажує результат (рис. 2.5).

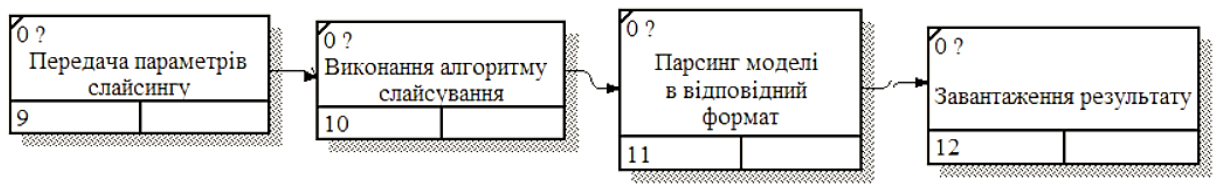


Рисунок 2.5 – Декомпозиція підпроцесу “Виконання слайсингу” в нотації IDEF3

2.4. Моделювання варіантів використання інформаційної технології слайсингу моделей для FDM 3D принтера

Подальша розробка потребує діаграми варіантів використання (Use Case Diagram) [17]. Дана діаграма відображає взаємодію акторів та варіантів використання. Діаграма варіантів використання описує, з точки зору особи, групу дій в системі, які призводять до конкретного результату. Розробка даного типу діаграм необхідна для потреби розуміння взаємодії між акторами та варіантами використання .

Для розроблюваної системи було визначено наступних акторів:

- Відвідувач – має доступ до основного функціоналу.
- Користувач з правом читання-запису – має доступ до основного функціоналу та можливість завантаження моделей для швидкого доступу.
- Адміністратор – має доступ до управління вільним місцем та зайнятим дисковим простором.
- WebGL, three.js – відображають завантажену модель.

Після визначення акторів в системі, формується перелік варіантів використання. Варіанти використання:

- Завантаження моделі.
- Перегляд моделі.
- Параметризація друку.
- Слайсинг.
- Збереження результатів слайсингу.
- Авторизація/Реєстрація.

На основі сформованих даних про варіанти використання та акторів було розроблено Use Case діаграму, яка представлена на рис. 2.6.



Рисунок 2.6 – Use Case Diagram

Відвідувачі можуть завантажувати модель, переглядати модель, зберігати результати слайсингу.

Користувачі можуть завантажувати модель, переглядати модель, зберігати результати слайсингу також можуть зберігати моделі для швидкого доступу.

Адміністратор може завантажувати модель, переглядати модель, зберігати результати слайсингу, зберігати моделі для швидкого доступу також управління записами користувачів та вільним місцем на диску.

WebGL разом з three.js забезпечують роботу з моделлю.

MySQL забезпечує авторизацію/реєстрацію користувачів.

2.5.Проектування моделі бази даних

При проектуванні бази даних було розроблено ER-діаграму, яка відображає таблиці та зв'язки між ними (рис. 2.7).

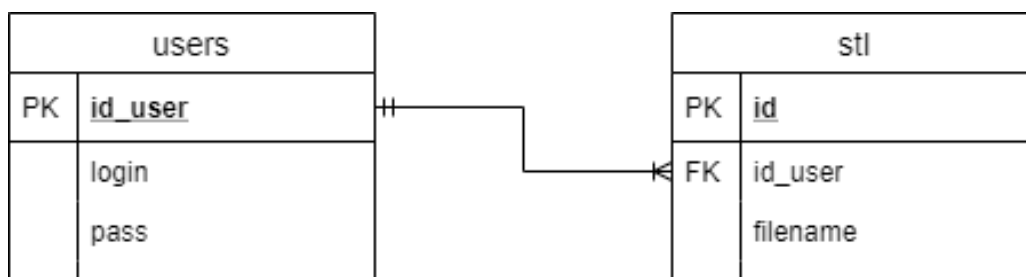


Рисунок 2.7 – ER-діаграма

База даних містить в собі таблицю “users” та таблицю “stl”. До першої таблиці заносяться дані користувача, в даному випадку логін та пароль, ця таблиця має зв'язок з таблицею “stl”, вона в свою чергу містить назви файлів, які користувач завантажив.

Інтерфейс користувача передбачає управління файлами, тобто користувач може видаляти файли, якщо вони непотрібні.

3 РОЗРОБКА WEB-ДОДАТКУ ДЛЯ НАЛАШТУВАННЯ МОДЕЛІ ПЕРЕД ДРУКОМ НА 3D-ПРИНТЕРІ

3.1 Архітектура web-додатку

Було використано дволанкову клієнт-серверну архітектуру, яка передбачає взаємодію двох програмних модулів — клієнтського та серверного, точніше модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта (рис. 3.1).

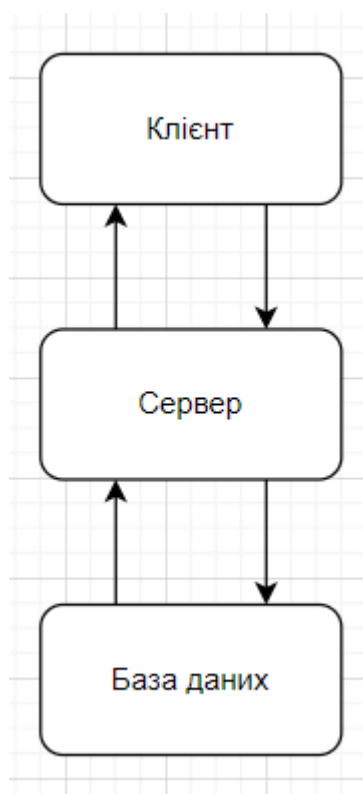


Рисунок 3.1 – Схема архітектури додатку

Компоненти web-додатку показані на рис. 3.2.

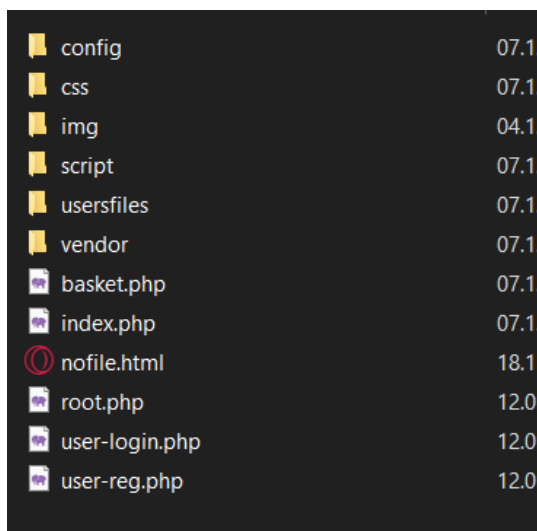


Рисунок 3.3 – Структура папок

Спочатку було реалізовано інтерфейс користувача, було використано Bootstrap, завдяки цьому створено зручний та зрозумілий інтерфейс [18].

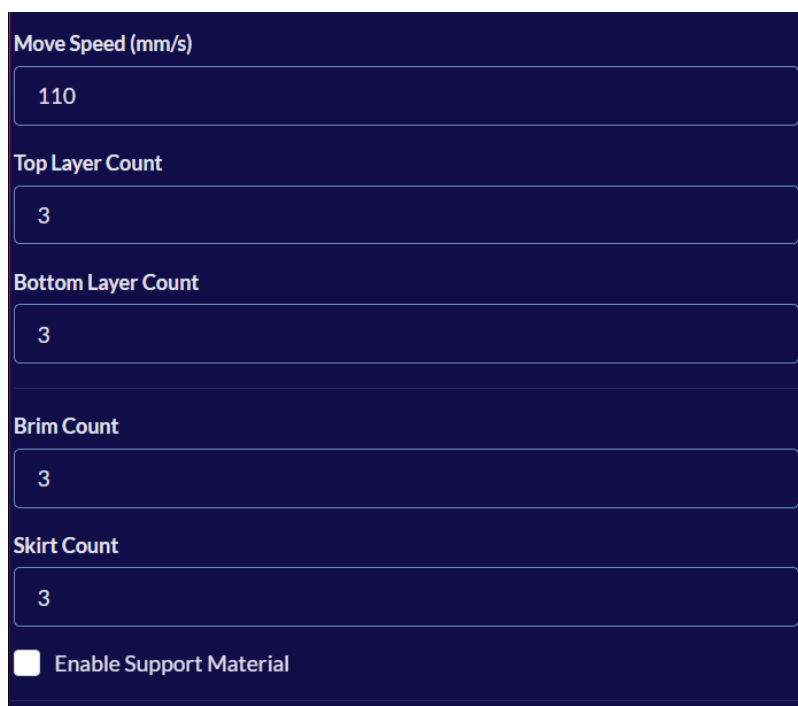
Приклад створення елементів “input”:

```
<div class="field">
  <label>Bed Depth (mm)</label>
  <input type="number" name="bed-depth" value="200">
</div>
```

Створивши елементи “input”, було додано клас з Bootstrap, після чого стилі підтягнулися стилі автоматично. У результаті було отримано поля, зображені на рис. 3.4.

Рисунок 3.4 – Створені поля

Таким чином було створено інші елементи панелі параметрів (рис. 3.5)



The image shows a dark-themed parameter panel with the following settings:

- Move Speed (mm/s): 110
- Top Layer Count: 3
- Bottom Layer Count: 3
- Brim Count: 3
- Skirt Count: 3
- Enable Support Material:

Рисунок 3.5 – Панель параметрів

Після створення панелі параметрів було реалізовано полотно canvas за допомогою JavaScript [19]. Фрагмент коду:

```
_canvasWidth = width * _pixelRatio;
_canvasHeight = height * _pixelRatio;

_canvas.width = _canvasWidth;
_canvas.height = _canvasHeight;

_canvasWidthHalf = Math.floor( _canvasWidth / 2 );
_canvasHeightHalf = Math.floor( _canvasHeight / 2 );

if ( updateStyle !== false ) {

    _canvas.style.width = width + 'px';
    _canvas.style.height = height + 'px';

}
```

Полотно завантажується та до нього підключаються бібліотека `three.js` також бібліотека `earcut.js`, файли реалізації відображення моделі [20], бібліотека перегляду моделі `stl_viewer.js`.

Фрагмент коду `stl_viewer.js` наведено нижче:

```
var material=new THREE.MeshLambertMaterial({ color:0x909090, wireframe: false,
vertexColors: model.color?THREE.NoColors:THREE.FaceColors}); //if model color is set,
ignores face colors set on the STL file itself (if any)
    if (!_this.is_ie) material.side = THREE.DoubleSide;
    if (!model.display) model.display="flat";
    _this.set_material_display(model.display, material, geo); //shading (aka display)
    model.mesh=new THREE.Mesh(geo, material);
    _this.set_model_custom_props(model); //position, color, scale
    _this.set_geo_minmax(model);
    model.mesh.model_id=model.id; //loop-back link (useful to detect clicks)
    _this.recalc_dims(model);
    _this.scene.add(model.mesh);
    _this.model_loaded(model.id);
    if (_this.model_loaded_callback)
        _this.model_loaded_callback(model.id);
}
```

`CanvasRenderer.js` відповідає за відображення полотна `canvas` та зв'язана з бібліотекою `three.js`, за допомогою такого зв'язку дозволяє відобразити завантажену користувачем модель на полотні [21]. Фрагмент коду бібліотеки `OrbitControls.js`:

```
THREE.SpriteCanvasMaterial.prototype = Object.create( THREE.Material.prototype );
THREE.SpriteCanvasMaterial.prototype.constructor = THREE.SpriteCanvasMaterial;
THREE.SpriteCanvasMaterial.prototype.isSpriteCanvasMaterial = true;
```

```
THREE.SpriteCanvasMaterial.prototype.clone = function () {
```

```
    var material = new THREE.SpriteCanvasMaterial();
```

```
    material.copy( this );
```

```
    material.color.copy( this.color );
```

```
    material.program = this.program;
```

```
    return material;
```

Зв'язок з бібліотекою `OrbitControls.js`, дозволяє управляти орбітами та повертати модель для того щоб роздивитися модель з усіх сторін [22].

Таким чином реалізація завантаження та відображення моделі закінчена.

Реалізація слайсингу моделі заключається в використанні та поєднанні бібліотек `earcut.js`.

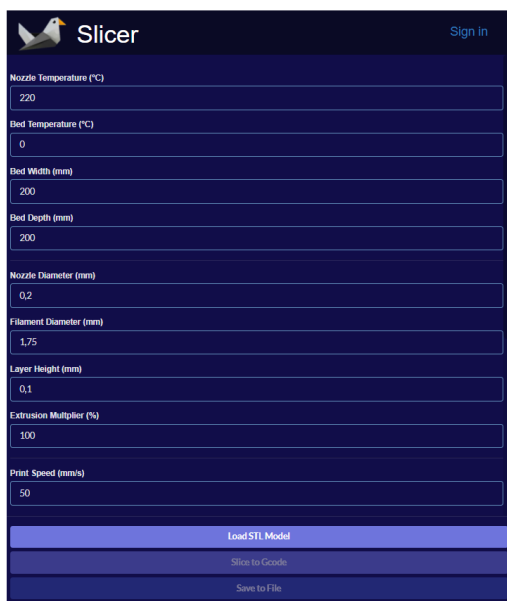
Бібліотека `earcut.js` реалізує модифікований алгоритм нарізання шарів, оптимізований хешуванням кривої z-порядку та розширений для обробки отворів, скручених багатокутників, не гарантує правильності тріангуляції, але намагається завжди дати прийнятні результати. Найшвидша і найменша бібліотека тріангуляції багатокутників JavaScript. Ця бібліотека достатньо швидка для тріангуляції в реальному часі, жертвуючи якістю тріангуляції заради швидкості та простоти, щоб обробляти більшість практичних наборів даних без збоїв [23].

`Earcut.js` може бути повільною для величезних складних фігур, але коли справа доходить до тріангуляції великої кількості фігур із відносно малою кількістю вершин у середньому.

3.3 Використання програмного додатку

У першу чергу додаток будуть використовувати спеціалісти, які займаються друком 3D моделей на 3D принтерах типу FDM. Основна роль додатку полягає в тому, щоб спеціаліст міг проінспектувати необхідну модель на наявність дефектів при друку, а також використати індивідуальні налаштування принтеру для більш точного та якісного аналізу [24].

Відкривши додаток, користувач побачить головну сторінку (рис. 3.6).



Load STL File
Choose STL file

Рисунок 3.6 – Головна сторінка web-додатку

Користувач завантажить необхідну модель, натиснувши на кнопку “Load STL Model” (рис. 3.7).

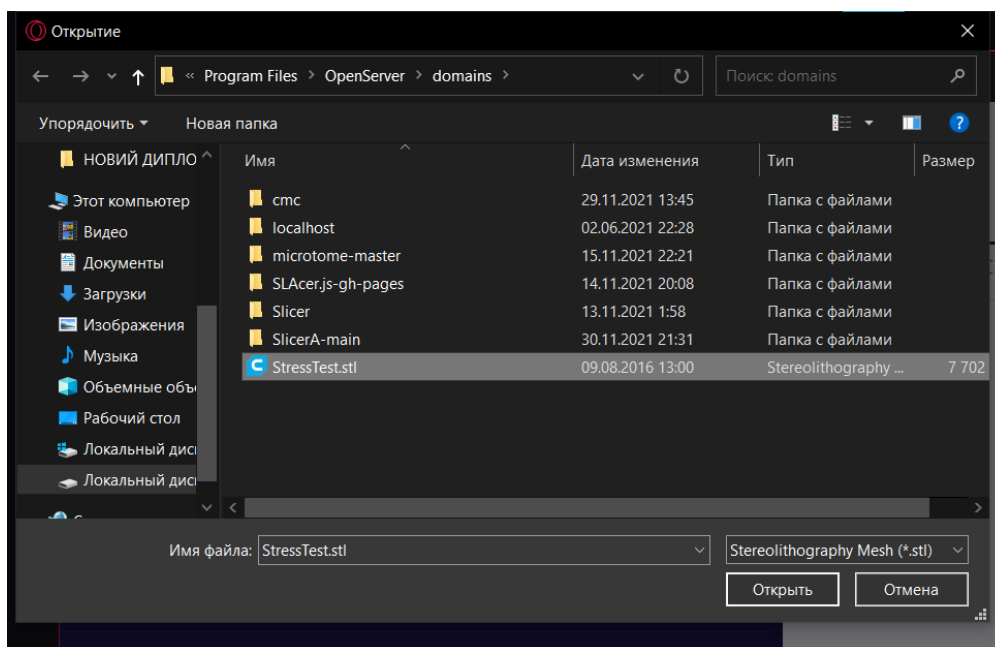


Рисунок 3.7 – Вибір необхідного файлу

Для прикладу було обрано тестову модель (рис. 3.8).

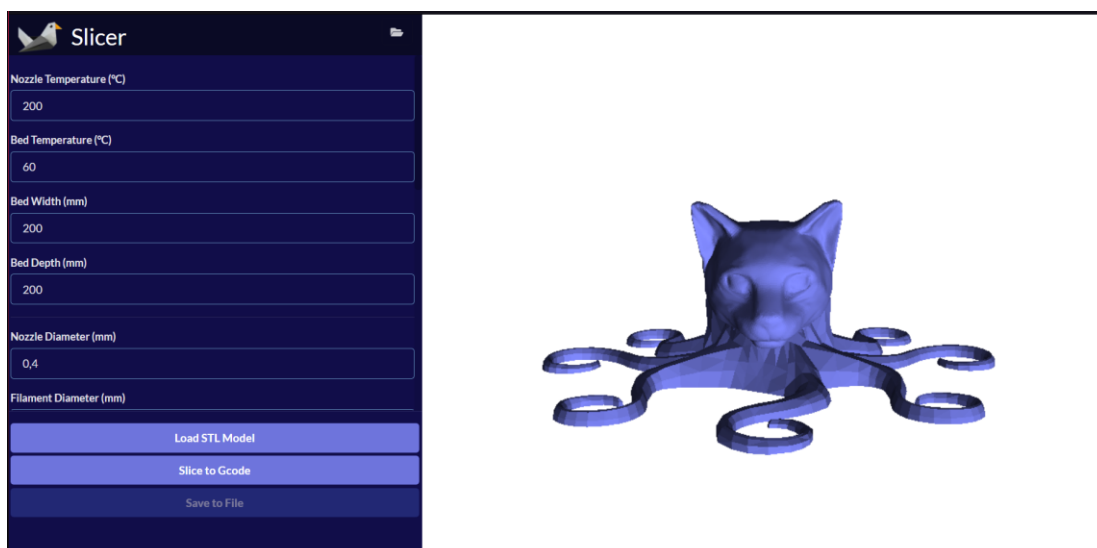


Рисунок 3.8 – Завантажена модель

Після завантаження необхідно встановити параметри друку відповідно до параметрів принтера, а також можна використати стандартні параметри (рис. 3.9).

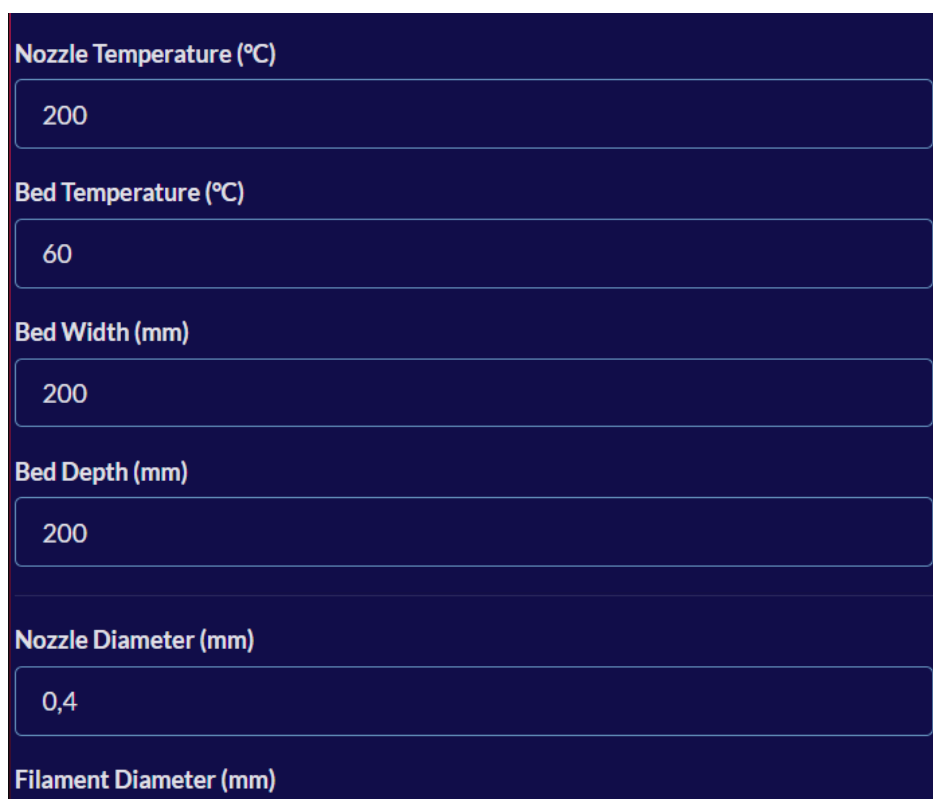


Рисунок 3.9 – Панелі налаштувань

Якщо користувачу потрібно згенерувати систему підтримок, необхідно лише встановити необхідний флаг та система буде згенерована (рис. 3.10).



Рисунок 3.10 – Модель після повороту

Наступним кроком встановлюються індивідуальні параметри принтеру, розмір та температурою друкуючої поверхні (рис.3.11).



Рисунок 3.11 – Параметри друкуючої поверхні

Пункт *Bed Width* відповідає налаштуванням розміру друкуючої поверхні. Пункт *Bed Depth* відповідає налаштуванням висоти друкуючої поверхні. Пункт *Bed Temperature* відповідає налаштуванням температури робочої поверхні [25].

Інші параметри, які необхідно змінити являються параметри розміру сопла, діаметр пластикової нитки, висота шару, відсоток екструзії матеріалу,

швидкості друку також параметри швидкості переміщення сопла, кількість останніх шарів, кількість перших шарів, ширину Brim, ширину Skirt [26] (рис. 3.12).

Nozzle Diameter (mm)	Move Speed (mm/s)
0,4	110
Filament Diameter (mm)	Top Layer Count
1,75	3
Layer Height (mm)	Bottom Layer Count
0,2	3
Extrusion Multiplier (%)	Brim Count
100	3
Print Speed (mm/s)	Skirt Count
50	3

Рисунок 3.12 – Параметри слайсингу

Після того, як користувач завантажив модель, вона відобразиться в canvas (рис. 3.13).

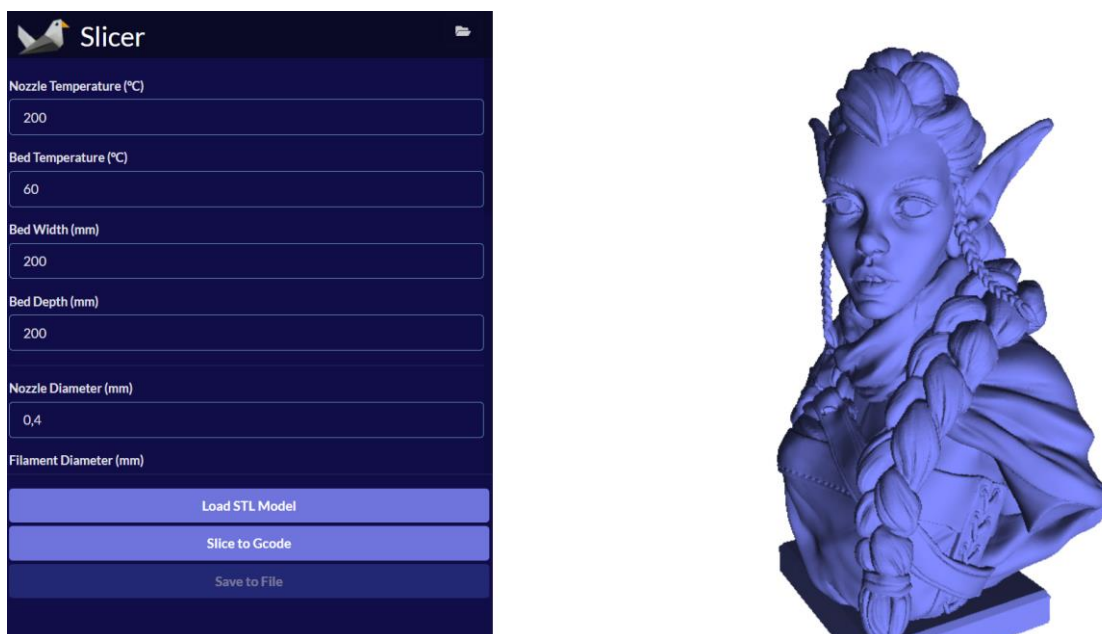


Рисунок 3.13 – Завантажена модель

Після завантаження моделі та встановлення параметрів слайсингу користувач натискає кнопку Slice to Gcode, в результаті відслайсована модель відображається користувачеві [27] (рис. 3.14).

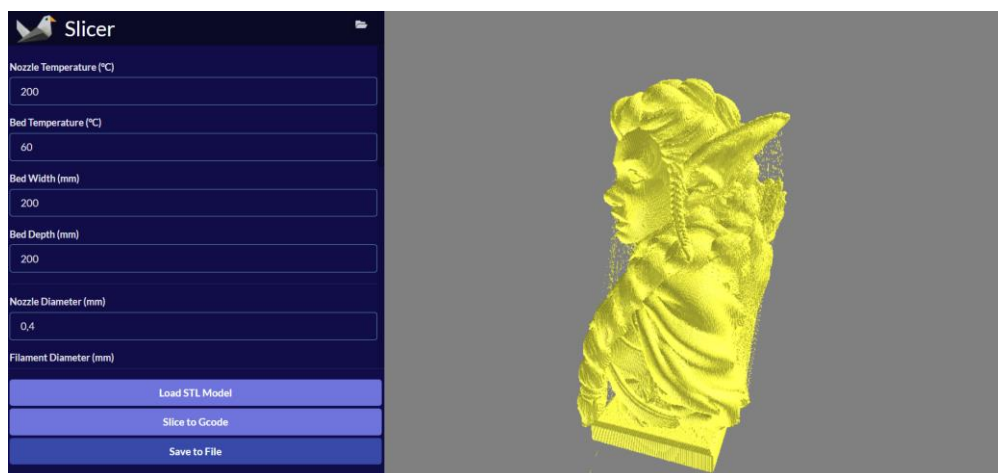


Рисунок 3.14 – Відображена відслайсована модель

Для того, щоб завантажити модель в форматі *.gcode, користувачу потрібно натиснути кнопку Save to File, файл буде збережено на локальний диск (рис. 3.15).

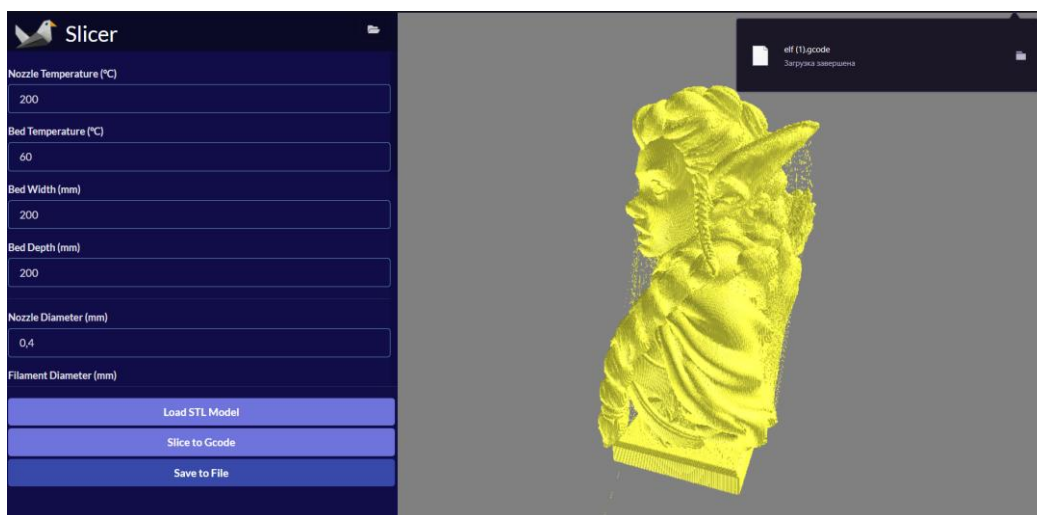


Рисунок 3.15 – Завантажений файл

Сам файл містить спеціальний код для контролера принтера, який в свою чергу розуміє та розпізнає, що та як йому необхідно надрукувати [29] (рис. 3.16).

```
elf (1).gcode – Блокнот
Файл  Правка  Формат  Вид  Справка
M107 ; disable fan;
$SET_INITIAL_TEMP
M104 S200 ; start heating hot end
M190 S60 ; heat and wait for bed
M109 S200 ; wait for hot end temperature
;START_GCODE
G1 Z5 F5000 ; lift nozzle
G92 E0 ; reset extrusion distance
;TYPE:SKIRT
G0 X104.75 Y81.62 Z0.10 F6600
G1 X105.84 Y85.37 F1500 E0.0649
G1 X105.89 Y87.06 E0.0931
G1 X105.67 Y88.02 E0.1094
```

Рисунок 3.16 – Вміст файлу *.gcode

Для швидкого доступу до моделей користувач може пройти реєстрацію та в особистому кабінеті завантажувати необхідні моделі (рис. 3.17).

Choose stl-file

Выберите файл Файл не выбран

Upload

id	filename	download	remove
1	calicat.stl	download	remove
10	zanahoria2.stl	download	remove
11	Deer1.stl	download	remove
12	astronaut.stl	download	remove
13	boobcube-v2.0.stl	download	remove
14	darthvader-1.stl	download	remove

Рисунок 3.17 – Особистий кабінет користувача

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи магістра було проведено аналіз предметної області, визначено основні вимоги до розробки, сформовано мету і задачі для реалізації проекту. Було розроблено інформаційну технологію слайсингу моделей для FDM 3D принтера, яка дозволяє без встановлення додатку провести певні маніпуляції з моделлю, встановити параметри свого принтеру та параметри слайсингу та в результаті отримати відслайсовану модель в необхідному форматі [30]. Було розглянуто додатки-аналоги.

Визначено функціональні вимоги, проведено планування, створено структуру робіт проекту, складено календарний план, проаналізовано ризики проекту. Також проведено структурно-функціональне моделювання, створено діаграму IDEF0, її декомпозицію та UML діаграму. Для реалізації було обрано: Visual Studio Code, HTML, Three.js, WebGL, earcut.js, CSS, OrbitControl.js, MySQL як базу даних для підтримки роботи веб-сайту [31].

Результатом кваліфікаційної роботи магістра є інформаційна технологія слайсингу моделей для FDM 3D принтера, реалізована у web-додатку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1 Support Settings [Електронний ресурс]. – Режим доступу: <https://all3dp.com/2/prus-slicer-support-settings-tree-supports/> – 01.12.2021р.
- 2 Setting up Support Material in FDM Slicers [Електронний ресурс]. – Режим доступу: <https://www.solidprint3d.co.uk/setting-up-support-material-in-fdm-slicers/> – 01.12.2021р.
- 3 Отличия, плюсы и минусы - технологии 3d печати [Електронний ресурс]. – Режим доступу: <https://getfab.ru/post/620/> – 01.12.2021р.
- 4 Slicer Method Comparison Using Open-source 3D Printer [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/323001996_Slicer_Method_Comparison_Using_Open-source_3D_Printer – 01.12.2021р.
- 5 Принцип работы 3d-принтеров [Електронний ресурс]. – Режим доступу: <http://3dpr.ru/printsip-raboty-3d-printera> – 01.12.2021р.
- 6 Возможности применения 3d-принтеров [Електронний ресурс]. – Режим доступу: <http://3dpr.ru/primenenie-3d-printerov> – 01.12.2021р.
- 7 Алгоритм слайсингу [Електронний ресурс]. – Режим доступу: [file:///D:/Downloads/Download/CAD_13\(1\)_2016_14-31.pdf](file:///D:/Downloads/Download/CAD_13(1)_2016_14-31.pdf) – 01.12.2021р.
- 8 Function representation based slicer for 3D printing [Електронний ресурс]. – Режим доступу: <https://www.cad-https://www.sciencedirect.com/science/article/abs/pii/S0167839618300268> – 01.12.2021р.
- 9 Identifying 3D Prints Using Slicing Parameters [Електронний ресурс]. – Режим доступу: file:///D:/Downloads/Download/CAD_14_2017_12-30.pdf – 01.12.2021р.

- 10 3D Printing: Slice Your Model [Электронный ресурс]. – Режим доступа: <https://libguides.ecu.edu/c.php?g=154937&p=6016326>– 01.12.2021р.
- 11 An Overview on 3D Printing Technology: Technological, Materials, and Applications [Электронный ресурс]. – Режим доступа: <https://reader.elsevier.com/reader/sd/pii/S2351978919308169?token=E593CF40313AC7A566EFE6B3E0B55FD4174A0BDDAC1C34FDCA3CE734F9C1CC45CAC862CBA6CDF780B5B5790D7E8A65DB&originRegion=eu-west-1&originCreation=20211031162331> –01.12.2021р.
- 12 What is 3D Printing? [Электронный ресурс]. – Режим доступа: <https://3dprinting.com/what-is-3d-printing/>– 01.12.2021р.
- 13 3D Printing: Applications in evolution and ecology [Электронный ресурс]. – Режим доступа: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6468079/>– 01.12.2021р.
- 14 The 3-D Printing Revolution [Электронный ресурс]. – Режим доступа: <https://hbr.org/2015/05/the-3-d-printing-revolution>– 01.12.2021р.
- 15 Comparing the accuracy of 3D slicer software in printed enduse parts [Электронный ресурс]. – Режим доступа: <https://iopscience.iop.org/article/10.1088/1757-899X/659/1/012082>– 01.12.2021р.
- 16 3D Slicer [Электронный ресурс]. – Режим доступа: <https://iopscience.iop.org/article/10.1088/1757-899X/342/1/012016/pdf> – 01.12.2021р
- 17 Three-dimensional Printing and 3D Slicer [Электронный ресурс]. – Режим доступа: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6026246/>– 01.12.2021р.
- 18 Алгоритм слайсингу [Электронный ресурс]. – Режим доступа: <https://github.com/substack/point-in-polygon/blob/master/index.js>– 01.12.2021р.
- 19 Отличия, плюсы и минусы - технологии 3d печати [Электронный ресурс]. – Режим доступа: <https://getfab.ru/post/620/> – 01.12.2021р.

- 20 3D-принтеры: зачем они нужны и как они работают [Электронный ресурс]. – Режим доступа: <https://zoom.cnews.ru/publication/item/54392> – 01.12.2021р.
- 21 Принцип работы 3d-принтеров [Электронный ресурс]. – Режим доступа: <http://3dpr.ru/printsip-raboty-3d-printera> – 01.12.2021р.
- 22 Возможности применения 3d-принтеров, Валентин Холмогоров, Дмитрий Горьков, БХВ-Петербург, 01.12.2021р.
- 23 Книга 3D-печать с нуля [Электронный ресурс]. – Режим доступа: <http://3dpr.ru/sravnenie-tehnologij-3d-pechati> – 01.12.2021р.
- 24 3D Slicer Software: How it Works and What to Expect [Электронный ресурс]. – Режим доступа: <https://www.solidprint3d.co.uk/3d-slicer-software-how-it-works-and-what-to-expect/> – 01.12.2021р.
- 25 Setting up Support Material in FDM Slicers [Электронный ресурс]. – Режим доступа: <https://www.solidprint3d.co.uk/setting-up-support-material-in-fdm-slicers/> – 01.12.2021р.
- 26 What is Slicing Software, and what does it do? [Электронный ресурс]. – Режим доступа: <https://www.additive-x.com/blog/what-is-slicing-software-and-what-does-it-do/> – 01.12.2021р.
- 27 3D Slicing Software – The Basics [Электронный ресурс]. – Режим доступа: <https://the3dbros.com/3d-slicing-software-the-basics/> – 01.12.2021р.
- 28 3D Slicer Settings for Beginners – 8 Things You Need to Know [Электронный ресурс]. – Режим доступа: <https://pinshape.com/blog/3d-slicer-settings-5-things-you-need-to-know-about-3d-printing-software/> – 01.12.2021р.
- 29 Возможности применения 3d-принтеров [Электронный ресурс]. – Режим <http://3dpr.ru/primenenie-3d-printerov> – 01.12.2021р.
- 30 A Quick Guide to Slicers [Электронный ресурс]. – Режим доступа: <https://buildbee.com/blog/article/a-quick-guide-to-slicers> – 18.04.2021р.

ДОДАТОК А. ПЛАНУВАННЯ РОБІТ

Ідентифікація ідеї проекту

В основі 3D - друку лежить принцип пошарового нанесення будь-якого матеріалу за допомогою лазерів та інших приладів.

Такий підхід відрізняє ряд найважливіших переваг перед традиційними техніками виробництва. Він створює об'єкти будь-якої геометрії без необхідності в численних допоміжних інструментах. Іншими словами, якщо підприємець вирішує змінити зовнішній вигляд товару, йому більше не потрібно міняти всю лінію виробництва. Потрібно внести лише кілька правок в комп'ютерну модель. Для того щоб це зробити необхідно мати спеціальне програмне забезпечення, яке допоможе створити тривимірну модель.

Метою проекту є розроблення інформаційної технології слайсингу моделей для FDM 3D принтера.

Деталізація мети методом SMART

S. Розробити інформаційну технологію яка перетворює 3D-модель в спеціальний формат, тобто перетворення моделі в машинний код.

M. Результатом виконання проекту, має бути інформаційна технологія, отримати доступ до якої можна ввівши адресу сайту до адресного рядку, в свою чергу, веб-додаток мусить бути розміщений на хостингу, пов'язані з ним бази даних(БД), форми та модулі мають бути вірно налаштовані та під'єднані.

A. Проект потребує розробника із знанням HTML, CSS, мови програмування JavaScript, Three.js, earcut.js, WebGL, Canvas, PHP, MySQL, яке використовується для розроблення даного додатку.

R. Мету реально досягнути, так як розроблення інформаційної технології здійснюється в Visual Studio Code, .

Т. Обмеженість в часі зумовлена рішенням замовника, щоб отримати продукт проекту згідно плану.

Описання фази розробки ІТ—проекту

Структурна декомпозиція робіт (work breakdown structure, WBS) - це ієрархічна структура робіт, побудована з метою логічного розподілу усіх робіт з виконання проекту і подана у графічному вигляді. Це сукупність декількох рівнів, кожний з яких формується в результаті розподілу роботи попереднього рівня на її складові. Елементом найнижчого рівня є група робіт, або так званий робочий пакет (work package).

WBS представлена на рис. А.1.

Планування структури організації, для впровадження готового проекту (OBS)

Наступним кроком розробки структури проекту є визначення організаційної структури (OBS) проекту.

Організаційна структура проекту (OBS) – є графічним відображенням учасників проекту (фізичних та юридичних осіб) та їхніх відповідальних осіб, залучених до реалізації проекту. На верхньому рівні OBS проекту знаходиться керівник та команда управління проектом; на наступному рівні – виконавці. Останнім рівнем OBS-структури є відповідальні особи виконавців. Це не обов’язково повинні бути керівники, а ті співробітники, яким доручено безпосередньо організувати і відповідати перед виконавцем за виконання конкретного елемента WBS-структури.

OBS структура представлена на рис. А.2.

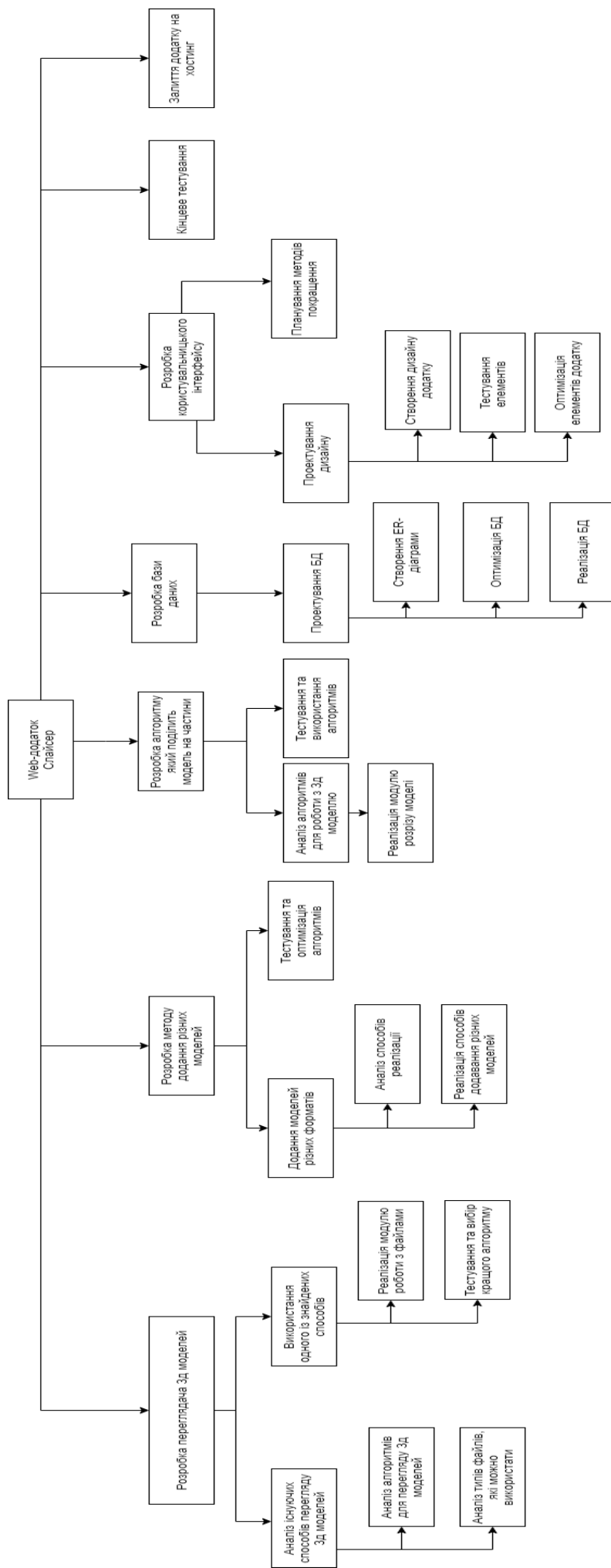


Рисунок А.1 – WBS діаграма

Побудова календарного графіку виконання ІТ—проекту

Діаграма Ганта (а Gantt Chart) - це візуальний спосіб відображення запланованих завдань. Горизонтальні графіки широко використовуються для планування проектів будь-яких розмірів в різних галузях і сферах. Це зручний спосіб показати, яка робота планується до виконання в певний день і час. Gantt Charts також допомагають командам і менеджерам проектів контролювати дати початку і закінчення будь-якого проекту.

Діаграма Ганта представлена на рис. А.3.

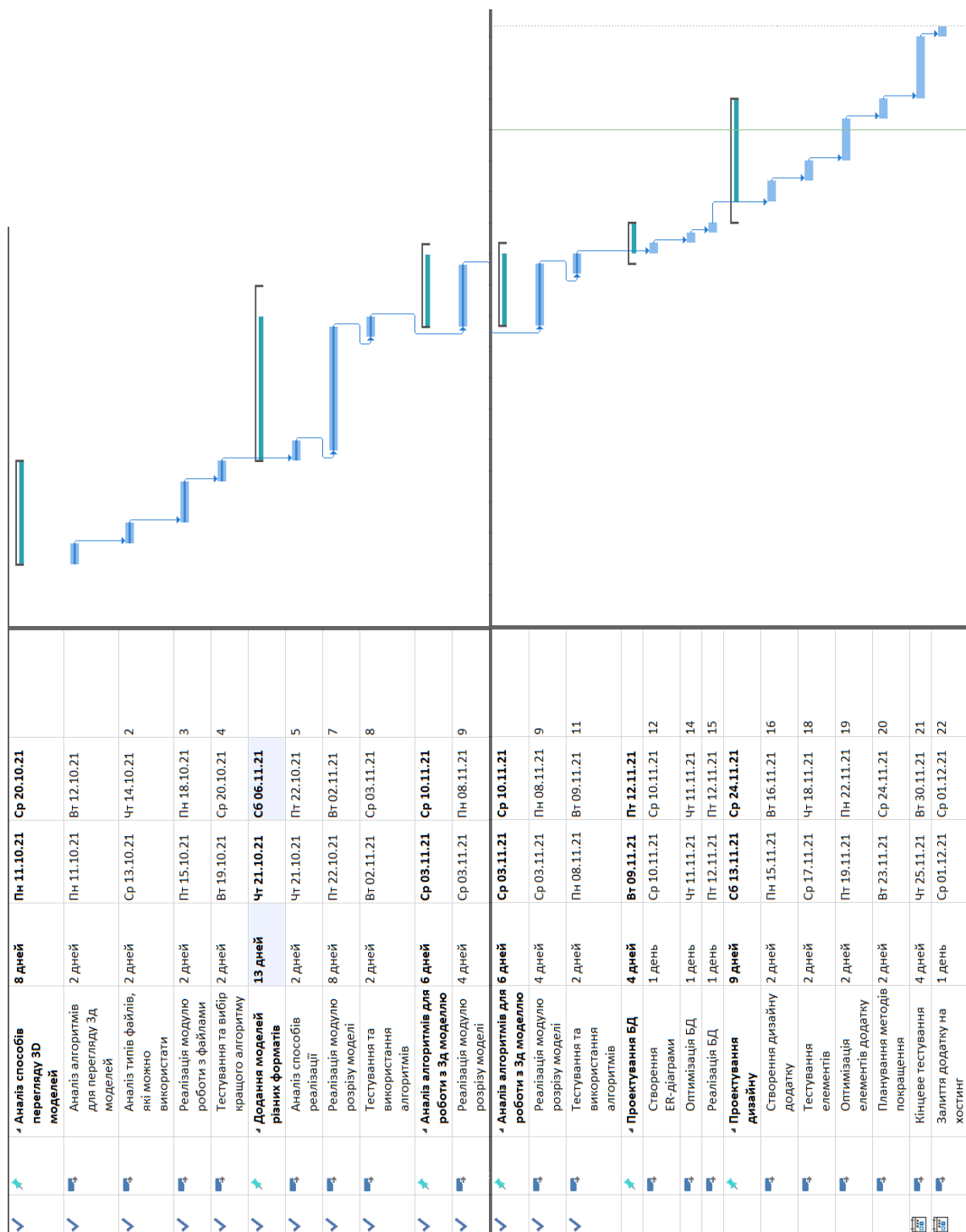


Рисунок А.3 – Діаграма Ганта

Ідентифікація ризиків

Ризик – це імовірна подія, яка у випадку своєї появи негативно або позитивно вплине на проект.

Процес управління ризиками включає наступні етапи:

1. ідентифікація

2. процес оцінювання ризиків, який включає в себе якісний, кількісний аналіз.
3. заходи реагування на ризики
4. моніторинг заходів і ризиків

Ризики проекту представлені у таблиці 1.1

Таблиця 1.1 – Реєстр ризиків

№	Опис ризику	Вплив	Ймовірність	РК	Пом'якшення
1	Проект не чітко визначений	4	2	С	Завершіть бізнес-обґрунтування, якщо ще не надано, та переконайтеся, що ціль чітко визначена в Статуті проекту та РІД.
2	Позапланова робота, яка буде виконана на пізній стадії проекту	4	3	С	Задokumentуйте всі припущення, зроблені під час планування, і повідомте менеджера проекту.
3	Тиск довільного скорочення тривалості завдання.	3	1	Н	Покажіть план проекту зацікавленим сторонам і поясніть, що скорочення часу виконання.
4	Низька залучення клієнтів	2	1	Н	Поясніть замовнику, його участь у проекті.
5	Важкі чи нездійсненні завдання	5	3	В	Оновіть персонал або замініть/відмініть завдання

Рівні критичності

- 1 Незначно
- 2 Низький
- 3 Середній
- 4 Високий

5 Критично

Таблиця 1.2 – Матриця ймовірності/впливу

5					
4					
3				2	5
2		4		1	
1			3		
Ймовірність / Вплив	1	2	3	4	5

ДОДАТОК Б

Index.html

```

<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="shortcut icon" type="image/jpg" href="img/small_icon.png"/>
    <title>SlicerA</title>
    <link rel="stylesheet" href="script/semantic/semantic.min.css">
    <script src="script/jquery.min.js"></script>
    <script src="script/semantic/semantic.min.js"></script>
    <script src="script/viewstl/build/stl_viewer.min.js"></script>
    <script src="script/gcode-viewer/gcode-viewer.min.js"></script>
    <script src="../script/ao_module.js"></script>
    <link rel="stylesheet" type="text/css" href="vendor/jquery-ui/jquery-
ui.min.css" />
    <link rel="stylesheet" type="text/css"
href="vendor/bootstrap/css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="vendor/bootstrap-
slider/css/bootstrap-slider.min.css" />
    <link rel="stylesheet" type="text/css" href="vendor/bootstrap-
colorpicker/css/bootstrap-colorpicker.min.css" />
    <link rel="stylesheet" type="text/css" href="css/main.css" />
    <style>
      body{
        background-color:white;
        position: absolute;
        top:0px;
        left:0px;
        height: 100%;
        width: 100%;
        padding-top: 14px;
        padding-left: 12px;
        padding-right: 12px;
        overflow: hidden;
      }

      .previewWindow{
        width: 100%;
        height: 100%;
        border: 0px solid transparent;

```

```

        background-color: #f7f7f7 !important;
    }

    .navi{
        width: 100%;
        height: 50px;
        background-color: #0a0a25;
        padding: 4px;
        padding-left: 12px;
        padding-right: 12px;
        color: #ffffff;
    }

    .optionbar{
        overflow-y: scroll;
        max-height: calc(100% - 230px);
    }

    .column{
        padding: 0px !important;
    }

</style>
</head>
<body>
    <div class="ui grid">
        <div class="six wide column">
            <div class="navi">
                
                <p style="pointer-events: none; user-select: none; font-size: 2em; padding-top: 3px; display: inline-block;">Slicer</p>
                <button class="ui basic right floated icon button"
onclick="selectFileToSlice();" style="color: rgba(255, 255, 255, 0.87) !important;" title="Open STL"><i class="folder open icon"></i></button>

                <ul class="nav navbar-nav navbar-right">
                    <?php
                    if (!isset($_SESSION["id_user"])) {
                        ?>
                        <li><a href="user-login.php" id="login_pop"><span
class="first-word">Sign in</a></li>
                    <?
                    };
            
```

```

        if (isset($_SESSION["admin"])) {
            ?>
            <li><a href="root.php"><span class="first-
word">Adminka</a></li>
            <?
        };
        if (isset($_SESSION["id_user"])) {
            ?>
            <li><a href="basket.php"><span class="first-
word">FileStore</a></li>
            <li><a href="config/out.php"><span class="first-
word">Logout</a></li>
            <?
        }
        ?>
    </ul>
</div>
<div id="optionForm" class="optionbar" style="padding-left: 4px;">
<br>
<form class="ui form" id="sliceConfig">
    <!-- Bed related configs -->
    <div class="field">
        <label>Nozzle Temperature (°C)</label>
        <input type="number" name="hot-end-temperature"
value="200">
    </div>
    <div class="field">
        <label>Bed Temperature (°C)</label>
        <input type="number" name="bed-temperature" value="60">
    </div>
    <div class="field">
        <label>Bed Width (mm)</label>
        <input type="number" name="bed-width" value="200">
    </div>
    <div class="field">
        <label>Bed Depth (mm)</label>
        <input type="number" name="bed-depth" value="200">
    </div>

    <div class="ui divider"></div>

    <!-- Extrusion related configs -->
    <div class="field">
        <label>Nozzle Diameter (mm)</label>

```

```

        <input type="number" unit="micrometer" name="extrusion-
width" min="0.1" step="0.1" value="0.4">
    </div>
    <div class="field">
        <label>Filament Diameter (mm)</label>
        <input type="number" unit="micrometer" name="filament-
diameter" min="0.1" step="0.05" value="1.75">
    </div>
    <div class="field">
        <label>Layer Height (mm)</label>
        <input type="number" unit="micrometer" name="layer-
thickness" min="0.1" step="0.05" value="0.2">
    </div>
    <div class="field">
        <label>Extrusion Multiplier (%)</label>
        <input type="number" name="extrusion-multiplier"
value="100">
    </div>

    <!-- Speed related settings -->
    <div class="ui divider"></div>
    <div class="field">
        <label>Print Speed (mm/s)</label>
        <input type="number" name="layer-speed" value="50">
    </div>
    <div class="field">
        <label>Move Speed (mm/s)</label>
        <input type="number" name="move-speed" value="110">
    </div>
    <div class="field">
        <label>Top Layer Count</label>
        <input type="number" name="number-top-layers" value="3">
    </div>
    <div class="field">
        <label>Bottom Layer Count</label>
        <input type="number" name="number-bottom-layers"
value="3">
    </div>

    <div class="ui divider"></div>
    <!-- Adhesion related configs-->
    <div class="field">
        <label>Brim Count</label>
        <input type="text" name="brim-count" value="3">
    </div>

```

```

<div class="field">
  <label>Skirt Count</label>
  <input type="text" name="skirt-count" value="3">
</div>

<div class="field">
  <div class="ui checkbox">
    <input type="checkbox" name="support-enabled" tabindex="0"
class="hidden">
    <label>Enable Support Material</label>
  </div>
</div>
</form>
</div>
<div class="ui divider" style="margin-top: 0px;"></div>
<div style="padding-right: 4px; padding-left: 4px;" >
  <button onclick="selectFileToSlice();" class="ui fluid button
actionbtn" style="background-color: #6e74dc; color: white;"> Load STL Model
</button>
  <button id="slicebutton" onclick="previewGcode(this);" class="ui
fluid disabled button actionbtn" style="background-color: #6e74dc; color: white;
margin-top: 4px;"> Slice to Gcode </button>
  <button id="savebutton" onclick="saveGcode();" class="ui fluid
disabled button actionbtn" style="background-color: #3849aa; color: white;
margin-top: 4px;"> Save to File</button>
</div>

</div>
<div id="preview" class="ten wide column">
  <iframe id="previewIframe" class="previewWindow"
src='./nofile.html'>

  </iframe>
</div>

<div id="savedDialog" class="ui basic modal">
  <div class="ui icon header">
    <i class="green checkmark icon"></i>
    Gcode Saved
  </div>
  <div class="content" align="center">
    <p>The gcode file has been saved to your file system</p>
  </div>
  <div class="actions">

```

```

        <div class="ui ok inverted button">
            OK
        </div>
    </div>
</div>
</div>

<script>
    let gcodeFilePath = "";
    let stlFilePath = "";
    let inputModelFilename = "";
    let currentSettings = { };
    let standaloneMode = false;
    let bufferedGcode = "";
    let standalone_gcodeRenderer = null;

    //Send a warning to user about the ao_module not found issue
    $(document).ready(function(){
        console.log("Hey, if you see the ao_module.js not found error above
this line, that means you are running this script in stand-alone mode which is
normal :)");
    });

    $('.ui.checkbox').checkbox();
    $(".ui.modal").modal();

    //Check if there are any input files
    if (typeof ao_module_loadInputFiles != 'undefined'){
        var inputFiles = ao_module_loadInputFiles();
        if (inputFiles != null){
            //We can only handle one file at a time
            var file = inputFiles[0];
            console.log(file);
            loadSTLFile(file.filename, file.filepath);
        }else{
            $("#previewIframe").attr("src", "./nofile.html");
        }

        //There is no place to show the author attributes :(
        $("#authorAttributes").hide();
    }else{

        standaloneMode = true;
        $("#previewIframe").attr("src", "./nofile.html");
    }
}

```

```

//Load all input field value from localStorage if exists
var lastSettings = localStorage.getItem("SlicerA-settings");
if (lastSettings == "" || lastSettings == null || lastSettings == undefined){
  //No setting exists. Build one from default settings
  $("input").each(function(){
    if ($(this).type != "checkbox"){
      currentSettings[$(this).attr("name")] = $(this).val();
    }
  });
}else{
  //Setting exists. Restore it
  currentSettings = JSON.parse(lastSettings);
  $("input").each(function(){
    var key = $(this).attr("name");
    if (currentSettings[key] != undefined){
      $(this).val(currentSettings[key]);
    }
  });
}

//Create event listener to remember all input change in localStorage
$("input").on("change", function(data){
  var newValue = $(this).val();
  currentSettings[$(this).attr("name")] = newValue;
  localStorage.setItem("SlicerA-settings",
JSON.stringify(currentSettings));
});

function handleSliceAndDownload(btn, callback=undefined){
  $(btn).addClass("loading");
  //Read the file as blob
  let reader = new FileReader();
  reader.onload = function() {
    //console.log(reader.result);
    var configs = generateConfigFromForm();
    var options =JSON.stringify(configs);

    $.ajax({
      url: "./sliced?options=" + options,
      method: "POST",
      data: reader.result,
      success: function(data){

```



```

        if (data.error !== undefined){
            alert(data.error);
        }else{
            if (callback !== undefined){
                callback(data);
            }else{
                console.log(data);
            }
        }
        $(btn).removeClass("loading");

        //console.log(data);
    }
})

};

reader.readAsDataURL(stlFilePath);
}

function downloadBufferedGcode(gcodeData){
    //OK! Download the return as gcode file
    var element = document.createElement('a');
    element.setAttribute('href', 'data:application/gcode;charset=utf-8,' +
encodeURIComponent(gcodeData));
    element.setAttribute('download',
stlFilePath.name.replace(".stl", ".gcode"));

    element.style.display = 'none';
    document.body.appendChild(element);

    element.click();

    document.body.removeChild(element);
}

function previewGcode(btn){
    if (standaloneMode == true){
        //Running in standalone mode. Slice and download the file
        handleSliceAndDownload(btn, function(gcode){
            //Update the global gcode buffer and ready to be downloaded
            bufferedGcode = gcode;

            //Render the gcode to preview

```

```

        var renderer = new gcodeViewer.GCodeRenderer(gcode,
$("##preview").width(), $("##preview").height(), new
gcodeViewer.Color(0x808080))
        renderer.colorizer = new gcodeViewer.SimpleColorizer(new
gcodeViewer.Color("#f0f043"))
        $("##preview").html("");
        $("##preview").append(renderer.element())
        renderer.render().then(() => {
            //Append flag to the viewer to mark it as auto resize canvas
            $("##preview").find("canvas").addClass("autoresize");

            //Allow gcode saving
            $("##savebutton").removeClass("disabled");

            //Set the renderer to global object
            standalone_gcodeRenderer = renderer;
        });
    });
    return;
}

//Generate a gcode at tmp folder and load it as preview
var configs = generateConfigFromForm();
if (stlFilePath == ""){
    alert("No STL file selected")
    return;
}
$("#btn").addClass("loading");
$.ajax({
    url: "./slice",
    method: "post",
    data: {options: JSON.stringify(configs), file: stlFilePath},
    success: function(data){
        if (data.error != undefined){
            alert(data.error);
        }else{
            //Removing the loading animation
            $("#btn).removeClass("loading");

            //Load the gcode viewer
            $("##previewIframe").attr("src", "");
            setTimeout(function(){

```

```

        var url =
`../SystemAO/utilities/gcodeViewer.html#[{"filepath"%3A"${encodeURIComponent(data)}"%2C"filename"%3A"Gcode Preview"}]`;
        $("#previewIframe").attr("src", url);
    }, 500);

    //Enable the save button
    $("#savebutton").removeClass("disabled");
    gcodeFilePath = data;
    }
    }
    });
}

```

```

function generateConfigFromForm(){
    var config = {};

```

```

    //Generate the config from the settings
    $("#sliceConfig").find("input").each(function(){
        var key = $(this).attr("name");
        var val = $(this).val();
        if ($(this).attr("unit") == "micrometer"){
            //x1000 for this value
            val = parseFloat(val) * 1000;
        }else if ($(this).attr('type') == "checkbox"){
            val = ($(this)[0].checked == true)
        }else{
            val = parseFloat(val);
        }
        config[key] = val;
    });

```

```

    //Manually inject some configs into the object
    config["initial-bed-temperature"] = config["bed-temperature"];
    config["initial-hot-end-temperature"] = config["hot-end-temperature"];
    config["initial-layer-speed"] = Math.min(config["layer-speed"]/2, 25);
    config["initial-layer-thickness"] = Math.min(config["layer-
thickness"]/2, 100);

```

```

    return config;
}

```

```

//Select a location to save the gcode
function saveGcode(){
    var outputFilename = "";

```

```

if (inputModelFilename == ""){
    outputFilename = "output.gcode";
}else{
    var tmp = inputModelFilename.split(".");
    tmp.pop();
    outputFilename = tmp.join(".") + ".gcode";
}

if (standaloneMode){
    //Use the gcode in buffer
    downloadBufferedGcode(bufferedGcode);
}else{
    //Open File selector at the same folder as the src stl
    var tmp = stlFilePath.split("/");
    tmp.pop();
    var defaultPath = tmp.join("/");
    ao_module_openFileSelector(handleFileSave, defaultPath,
"new",false, {
        defaultName: outputFilename
    });
}

}

function handleFileSave(filedata){
    for (var i=0; i < filedata.length; i++){
        var filename = filedata[i].filename;
        var filepath = filedata[i].filepath;

        $.ajax({
            url: "saveGcode",
            data: {src: gcodeFilePath, dest: filepath},
            success: function(data){
                console.log(data);
                $("#savedDialog").modal("show");
            }
        })
    }
}

function loadSTLFile(filename, filepath){

```

```

    var targetURL =
`../SystemAO/utilities/stlViewer.html#[{"filepath"%3A"${encodeURIComponent
(filepath)}"%2C"filename"%3A"${filename}"}]`;
    $("#previewIframe").attr("src", targetURL);
    stlFilePath = filepath;
    $("#slicebutton").removeClass("disabled");
    $("#savebutton").addClass("disabled");
}

//Loading STL file
function standaloneLoadSTLFile(fileObject){
    $("#preview").html(`);

    //Start reading the stl file
    const reader = new FileReader();
    reader.addEventListener("load", function () {
        //Read completed
        var stl_viewer=new
StlViewer(document.getElementById("preview"), {
            models: [{id: 0, filename: reader.result, color: "#757de8"}],
            canvas_height: window.innerHeight,
            canvas_width: "100%"
        });

        //Update the attributes for slicing
        stlFilePath = fileObject;
        $("#slicebutton").removeClass("disabled");

    }, false);

    if (fileObject) {
        reader.readAsDataURL(fileObject);
    }

    console.log(fileObject);
}

function selectFileToSlice(){
    if (standaloneMode){
        var input = document.createElement('input');
        input.type = "file";
        input.multiple = false;
        input.accept = ".stl";
        input.onchange = e => {
            var files = e.target.files;

```

```

        standaloneLoadSTLFile(files[0]);
    }
    input.click();
}else{
    ao_module_openFileSelector(fileSelected, "", "file",false, option = {
        filter: ["stl"]
    });
}
}
function fileSelected(filedata){
    $("#previewIframe").attr("src", "");
    for (var i=0; i < filedata.length; i++){
        var filename = filedata[i].filename;
        var filepath = filedata[i].filepath;
        setTimeout(function(){
            loadSTLFile(filename, filepath);
        }, 500);
        inputModelFilename = filename;
    }
}
function handleWindowResize(){
    $("iframe").css("height", window.innerHeight);
    $("#optionForm").css("height",window.innerHeight - 200);
    if ($("#preview").find(".autoresize").length > 0){
        var targetCanvas = $("#preview").find(".autoresize");
        $(targetCanvas).css({
            width: $("#preview").width(),
            height: $("#preview").height()
        });
        if (standalone_gcodeRenderer != null){
            standalone_gcodeRenderer.resize($("#preview").width(),
$("#preview").height());
        }
    }
}
handleWindowResize();
$(window).on("resize", function(){
    handleWindowResize();
})
</script>
</body>
</html>

```