1. **მარკაპის ამოცანა**

<https://www.figma.com/file/w3zfXj2lBSb7KuZ567sUos/Enagramm?node-id=161%3A352&t=cdiEeUgh77SO2x82-0>

მისამართზე მდებარე პროექტში არსებული Desktop-9 გვერდის დიზაინის მიხედვით react.js და сss-ზე გვერდის მარკაპის აწყობა

1. **JavaScript ამოცანა**

**კლასის აღწერა:** Javascript კლასის შექმნა, რომელსაც ექნება text ველი, start() მეთოდი და onresult მოვლენა (event).

**მუშაობის პრინციპი:** მას შემდეგ რაც ამ კლასის მომხმარებელი შექმნის ამავე კლასის ობიექტს (ე.წ. instance-ს), text-ველს მიანიჭებს რაღაც ტექსტს და გამოიძახებს start() მეთოდს, ობიექტმა უნდა დაიწყოს მინიჭებული ტექსტის დაჭრა სასვენ ნიშნებზე/space-ზე შემეგი პრინციპით: თუ ტექსტის სიგრძე არის 150-ზე მეტი, მაშინ 150-ე და 230-ე სიმბოლოების შუალედში არსებულ სასვენ ნიშნებზე გაჭრას, შემდეგი პრიორიტეტით:

წერტილი, ძახილის ნიშანი, კითხვის ნიშანი, წერტილ-მძიმე, ძმიმე, space. ციკლი უნდა მიდიოდეს სანამ ტექსტი არ დამთავრდება, რომელიც შეიძლება იყოს ძალიან გრძელი მაგ: 10000 სიმბოლოიანი.

ტექსტების მოჭრის შედეგად მიღებული ყოველი ნაწილი, მომდევნო მაგალითში აღნიშნული როგორც sentence, ასინქრონულად გააგზავნოს ბექში (სერვერის მხარეს) დასამუშავებლად შემდეგნაირად:

var model = {

Language: 'ka’,

Speed: 100,

Text: sentence,

Voice: 0,

IterationCount: 2

};

$.ajax({

type: 'POST',

url: 'https://enagramm.com/Tools/SythesizeTextDAL',

dataType: 'json',

async: true,

contentType: "application/json; charset=utf-8",

data: JSON.stringify(model),

success: function (result) {

var sourceUrl = result.AudioFilePath;

},

error: function (xhr, status, error) {

alert(error);

}

});

სერვერიდან პასუხის ყოველი მიღება, რომელიც მაგალითში მოცემულია როგორც: var sourceUrl = result.AudioFilePath უნდა იყოს onresult მოვლენის წარმოქმნა, რომლის დამუშავების(handler-ის) პარამეტრად უნდა ვიღებდეთ სერვერის მხრიდან დაბრუნებულ მონაცემს.

კლასის გამოყენების მაგალითი:

MyClass = new MyClass();

MyClass.text='გრძელი ტექსტი დასამუშავებლად. სასვენი ნიშნებით და სხვა სიმბოლოებით’

MyClass.start()

MyClass.onresult = function (result) {

var MyResult = result.sourceUrl;

}