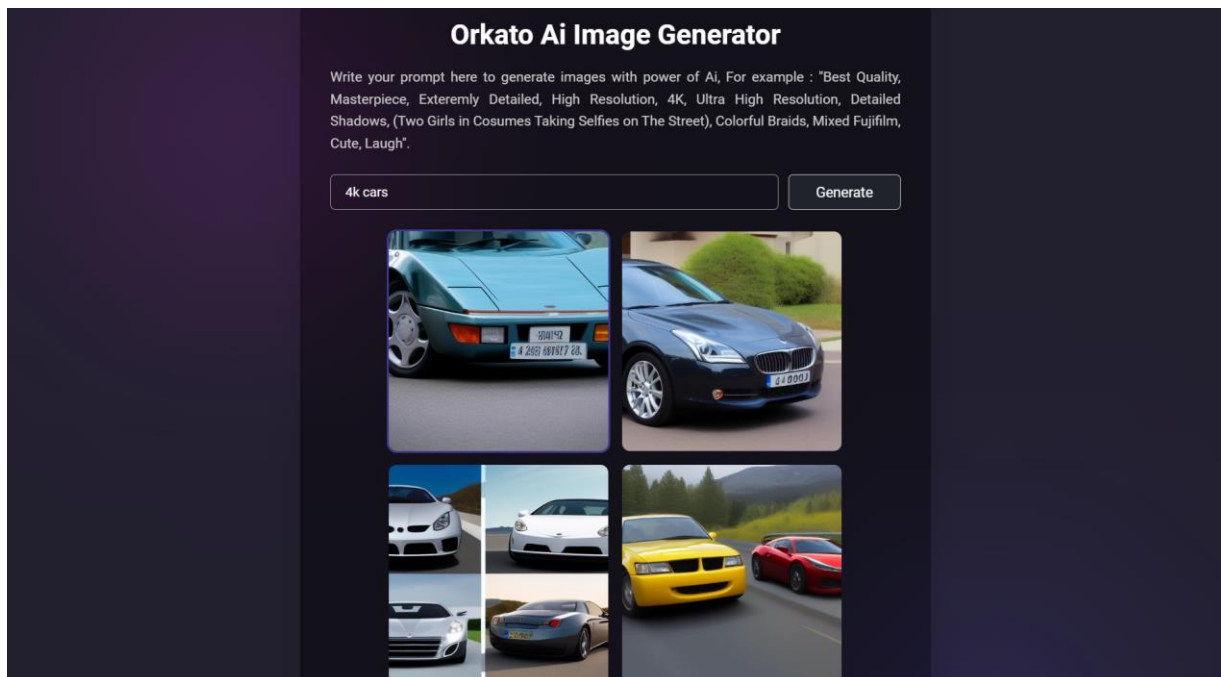
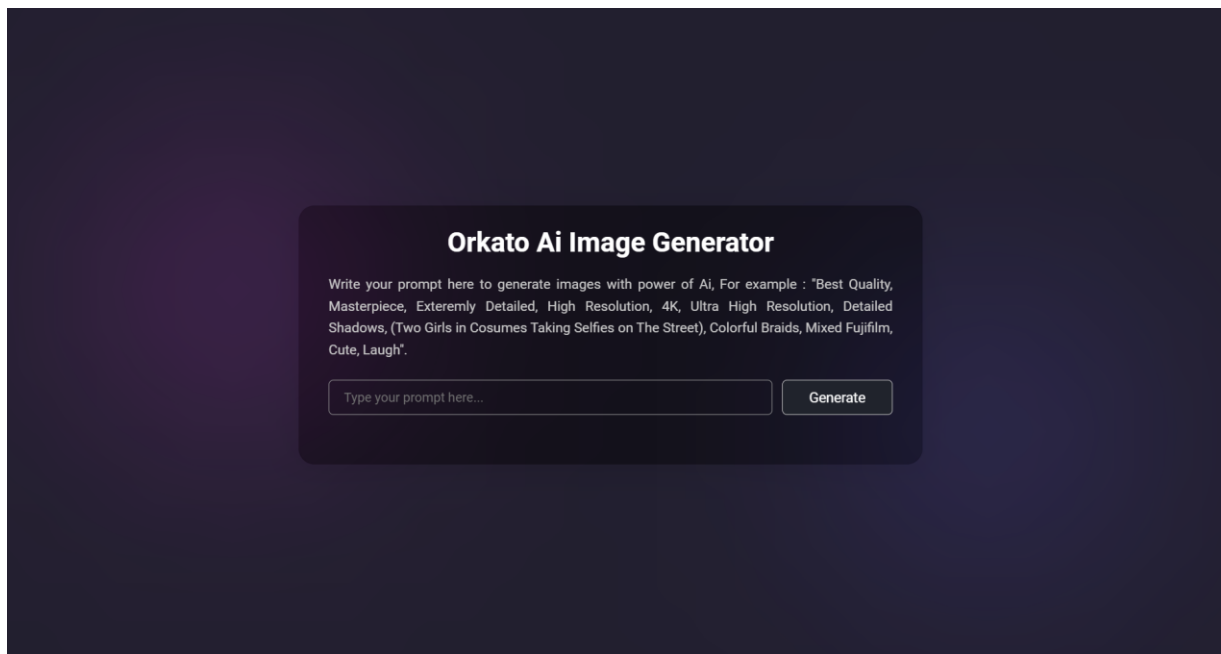
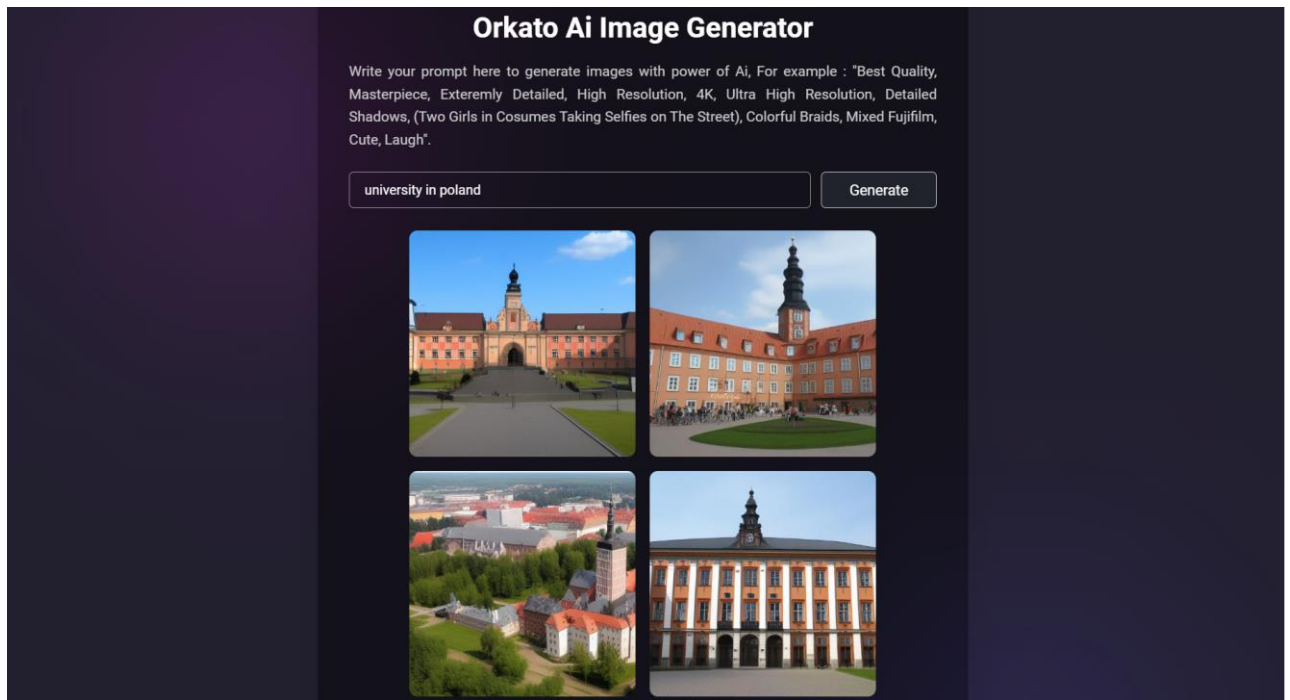


Ai Image Generator Website

Our website is an AI image generator that turns text descriptions into pictures. Users can easily create personalized images by typing what they want, and the advanced AI technology transforms their ideas into visually appealing artworks.





HTML Part:

1. `<!DOCTYPE html>`: Declares the document type and version of HTML being used.
2. `<html lang="en">`: Defines the root element of the HTML document, specifying the language as English.
3. `<head>`: Contains metadata about the document, such as character set, viewport settings, external stylesheets, and the document title.
 - `<meta charset="UTF-8">`: Sets the character set to UTF-8.
 - `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Configures the viewport settings for responsive design.
 - `<link rel="stylesheet" href="style.css">`: Links an external stylesheet named "style.css" to the HTML document.
 - `<title>Orkan Ai Image Generator | AsmrProg</title>`: Sets the title of the webpage.
4. `<body>`: Contains the content of the webpage.
 - `<div class="container">`: Defines a container for the main content of the page.
 - `<h1>`: Displays the main heading of the page, "Orkato Ai Image Generator."
 - `<p>`: Provides a paragraph of text instructing the user to enter a prompt to generate images using AI.
 - `<form class="gen-form">`: Defines a form with a text input and a button for generating images.
 - `<input type="text" id="user-prompt" , placeholder="Type your prompt here..."`

- `autocomplete="off">`: Provides a text input for the user to enter their prompt.
- `<button type="button" id="generate">Generate</button>`: A button that triggers the image generation process.
- `<div class="result">`: Contains the result section, including a loading message and an image grid.
- `<div id="loading">Generating...</div>`: Displays a loading message while the images are being generated.
- `<div id="image-grid"></div>`: Placeholder for the generated images.
- `<script src="script.js"></script>`: Links an external JavaScript file named "script.js" to the HTML document.

CSS Part:

Import Font:

Imports the "Roboto" font from Google Fonts.

Global Styles:

Resets margin, padding, and sets box-sizing to "border-box" for all elements.

Applies the "Roboto" font as the default font for the entire page.

Body:

Sets the background color to a dark shade.

Centers content both vertically and horizontally.

Container:

Styles the main container for content.

Gives it a background with a semi-transparent black overlay.

Adds padding, border-radius, and box-shadow for a card-like appearance.

Uses backdrop-filter for a blurred effect.

Container Pseudo-elements:

Creates two pseudo-elements before and after the container with a blurred background for decorative purposes.

Heading (h1):

Sets the font size, weight, and alignment for the main heading.

Paragraph (p):

Defines styles for paragraphs, including font size, color, line height, and text alignment.

Form (gen-form):

Styles the form, including margin, flex layout, and gap between elements.

Input:

Styles the text input with a transparent background, white text, and a border that changes on hover and focus.

Button:

Styles the button with specific width, height, color, and border properties.

Changes background color and border on hover.

Loading Message (#loading):

Styles the loading message with a font size and margin.

Disabled Button:

Styles the disabled button with different colors and cursor style.

Result Section:

Sets styles for the result section, including margin, width, padding, and flex layout.

Image Grid (#image-grid):

Styles the grid for displaying images with columns, gap, and maximum width.

Image in Grid:

Sets styles for images in the grid, including maximum width, border-radius, and a border that changes on hover.

JS Part:

API Key and Configuration:

`const apiKey = "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";` Holds the API key for authentication.

`const maxImages = 4;` Defines the number of images to generate for each prompt.

Random Number Generator:

`function getRandomNumber(min, max) {...}` Generates a random number between the specified min and max (inclusive).

Button State Functions:

`function disableGenerateButton() {...}` Disables the "Generate" button during image generation.

`function enableGenerateButton() {...}` Enables the "Generate" button after image generation.

Clear Image Grid Function:

`function clearImageGrid() {...}` Clears the content of the image grid.

Image Generation Function:

async function generateImages(input) {...}: Disables the button, clears the image grid, and displays a loading message.

Generates a set number of images by making asynchronous API calls.

Handles API response, creating image elements, and appending them to the image grid.

Enables the button, hides the loading message, and resets the selected image number.

Event Listener for Generate Button:

document.getElementById("generate").addEventListener('click', () => {...}: Listens for a click on the "Generate" button and triggers the generateImages function with the user's input.

Image Download Function:

function downloadImage(imgUrl, imageNumber) {...}: Creates a link to download the clicked image with a filename based on its position in the set.