Web Authoring 

Final Brief

### Student

Gabriela Prado

Software 1

25/03/2025

[**Introduction**](#_q7tphuu8ff6i) **2**

[**Target Market and Site Objectives**](#_fwmratgszn3n) **3**

[Target Market](#_voeona3mi7as) 3

[● Schools/Academies:](#_782gsqwrwg5w) 3

[● Toys/Kids Product Companies:](#_yp95e2k0if16) 3

[● Parents/Tutors:](#_r1osfijhzjfu) 3

[Site Objectives](#_yp2fhmrsvj3b) 4

[**Site Map**](#_by7rt8sfdwlp) **5**

[**Initial Sketches**](#_27y2kiugpmxt) **6**

[Wireframes](#_ywbicf7zs81t) 6

[**Tools for Website Creation and Maintenance**](#_z073dz43fww7) **7**

[**User Interface Proporse**](#_u95o5kzfd5uw) **8**

[Color Palette](#_gqwrad8ecsry) 8

[Home Page](#_o79ih68hgupf) 8

[Games Page](#_4nsk5b2z36j8) 9

[Services Page](#_jup30gqpufp) 9

[Contact Page](#_tb7qn2bnknx) 10

[**Upload the Site to a Web Server**](#_j678caqkztdi) **10**

[**Test WebSite Functionality**](#_rnbw5d9d7jwh) **11**

[Navigation bar](#_2yhibtwnao29) 11

[Games Showcase](#_bndedis7rcy2) 12

[Contact Form](#_hol3jwu7s7yv) 13

[**Issues Encountered**](#_l1009adri26n) **13**

[1.Mobile Side Menu](#_6ua8a8a5szr2) 13

[2.Images issue](#_vay37bcexstm) 14

[**Future Planning**](#_oq7a0qcevqla) **14**

[Updating Text:](#_d1q9dev3rh7h) 14

[Creating New Pages](#_kttjyrijt8rw) 15

[Inserting Images](#_bfc3mlerctqv) 15

[Creating Links](#_cejge41t353k) 15

[Uploading Updated Content](#_u05v0g4b86zu) 16

# Introduction

StinkyCat Studio is an independent game development studio specializing in creating educational and simulation games for children and young audiences. The website highlights the company's projects, providing detailed information about the services it offers while presenting their vision and company description.

This project uses Express for server-side application logic, EJS for templating, and MySQL for database operations(to keep track of contact form). Additionally, it uses middleware like CORS for cross-origin requests and dotenv for environment variable management.

<https://github.com/PradoGabriela/StinkyCatStudio_WebPrototype.git>

# Target Market and Site Objectives

## Target Market

The website is intended for three primary audiences:

### Schools/Academies:

* + **Demographics:** Schools and academies teaching computer science, robotics, and game design.
  + **Location:** Across the EU and UK, especially those catering to students from middle- or high-income families.
  + **Technology Usage:** Low-performance PCs.
  + **Pain Points:** Lack of information for specific age groups of kids.

### Toys/Kids Product Companies:

* + **Demographics:** Businesses specializing in products for children, potentially integrating games into their offerings.
  + **Technology Usage:** Mobile and PC support.
  + **Pain Points:** Difficulty in finding clear information for getting a quote.

### Parents/Tutors:

* + **Demographics:** Parents or guardians looking for educational or simulation games for their children.
  + **Technology Usage:** Mobile and PC support.
  + **Pain Points:** Overload of information on websites, making navigation difficult.

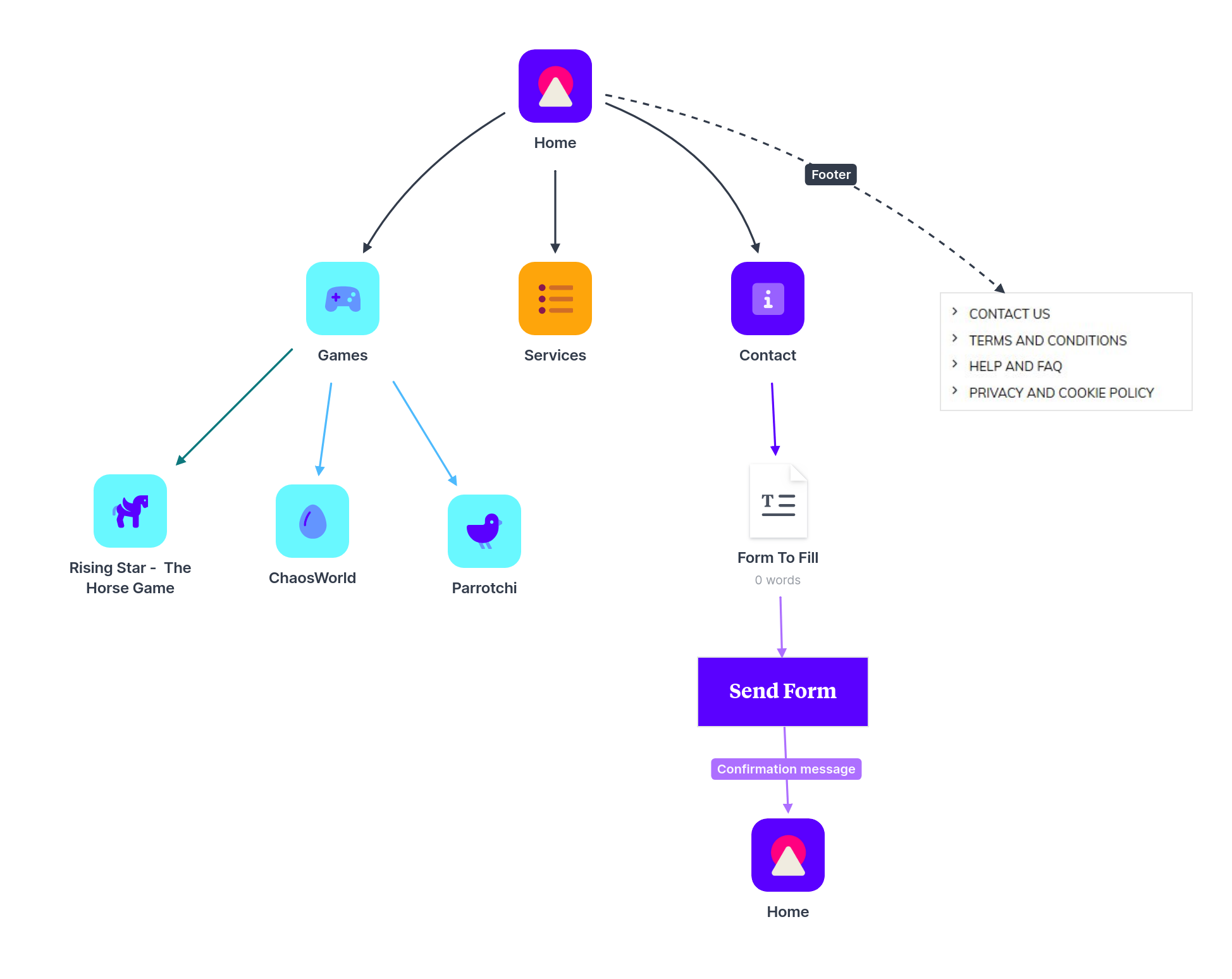
## Site Objectives

* Provide **clear and concise information** about services, products, and examples.
* Ensure an **easy navigation experience** for parents.
* Offer **contact forms and quotation options** for businesses and schools.
* Include **learning games and interactive content** to attract schools and academies.
* Ensure **compatibility across various devices** (PC and mobile).

This site aims to bridge the gap between educational institutions, toy companies, and parents by offering an intuitive and informative platform.

# 

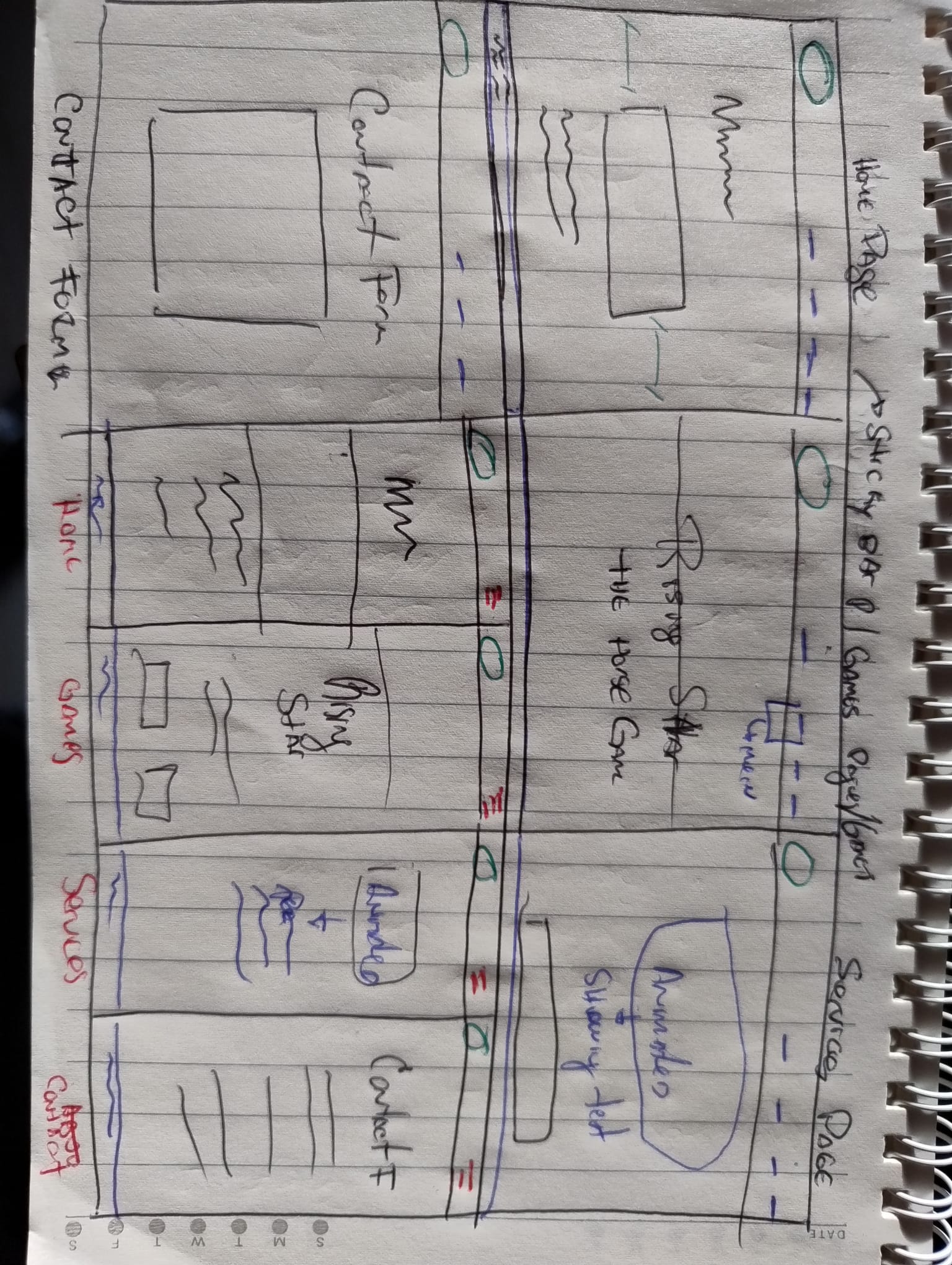
# Site Map



# Initial Sketches

The following sketches outline initial design concepts for the website, featuring key pages such as a **home, games, services and contact**, navigation links, game listings, and the contact section include a simple form.

### Wireframes



|  |  |
| --- | --- |
|  |  |

# Tools for Website Creation and Maintenance

**Visual Studio Code** as a code editor.

**Canva** as a graphic design tool.

**JavaScript, HTML and CSS.**

**node js** -and its dependencies: cors, dotenv, ejs, env, express, fs, mysql.

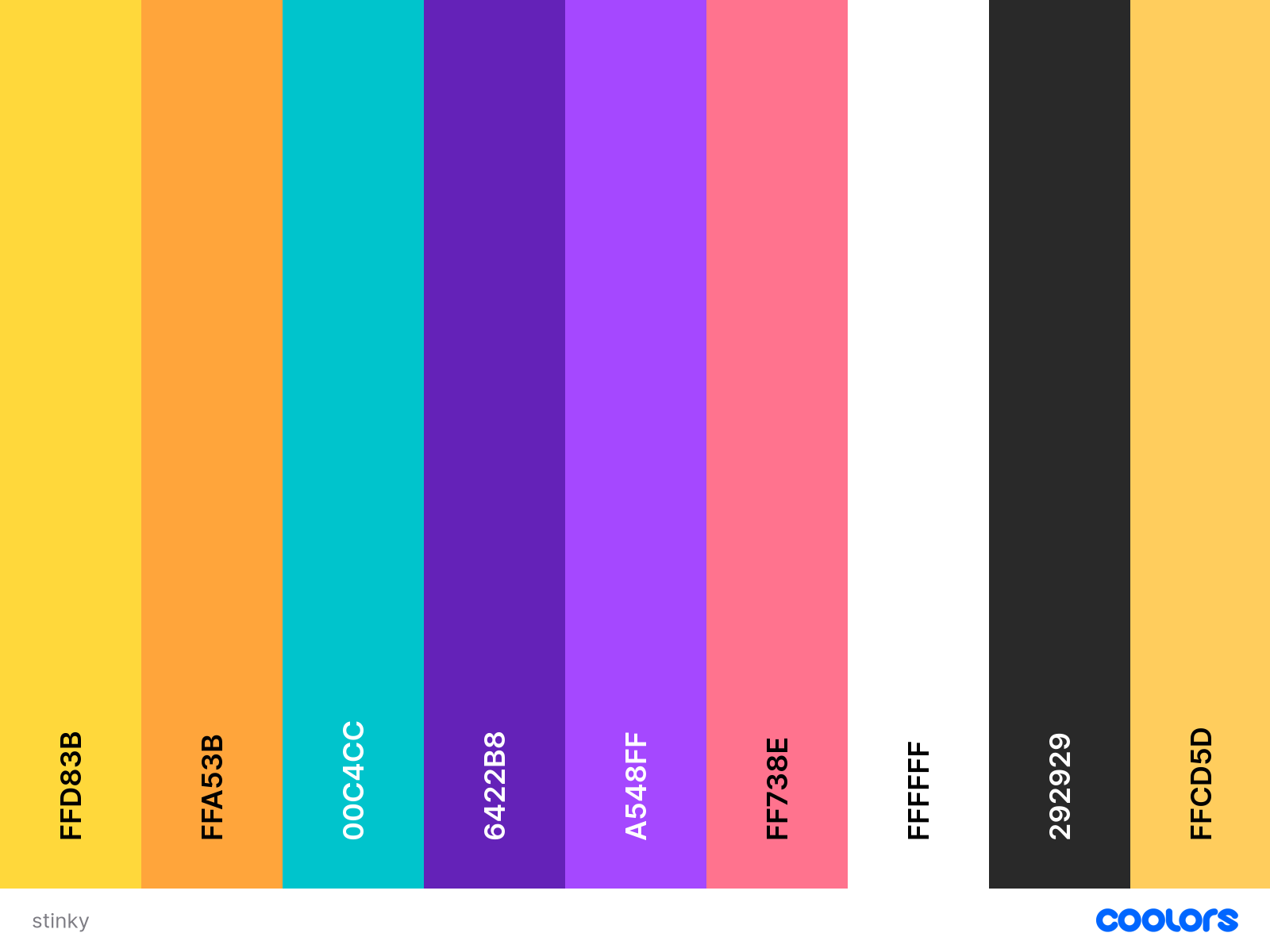
**MySql** as database

Google fonts, Font Awesome and GitHub.

# User Interface Proporse

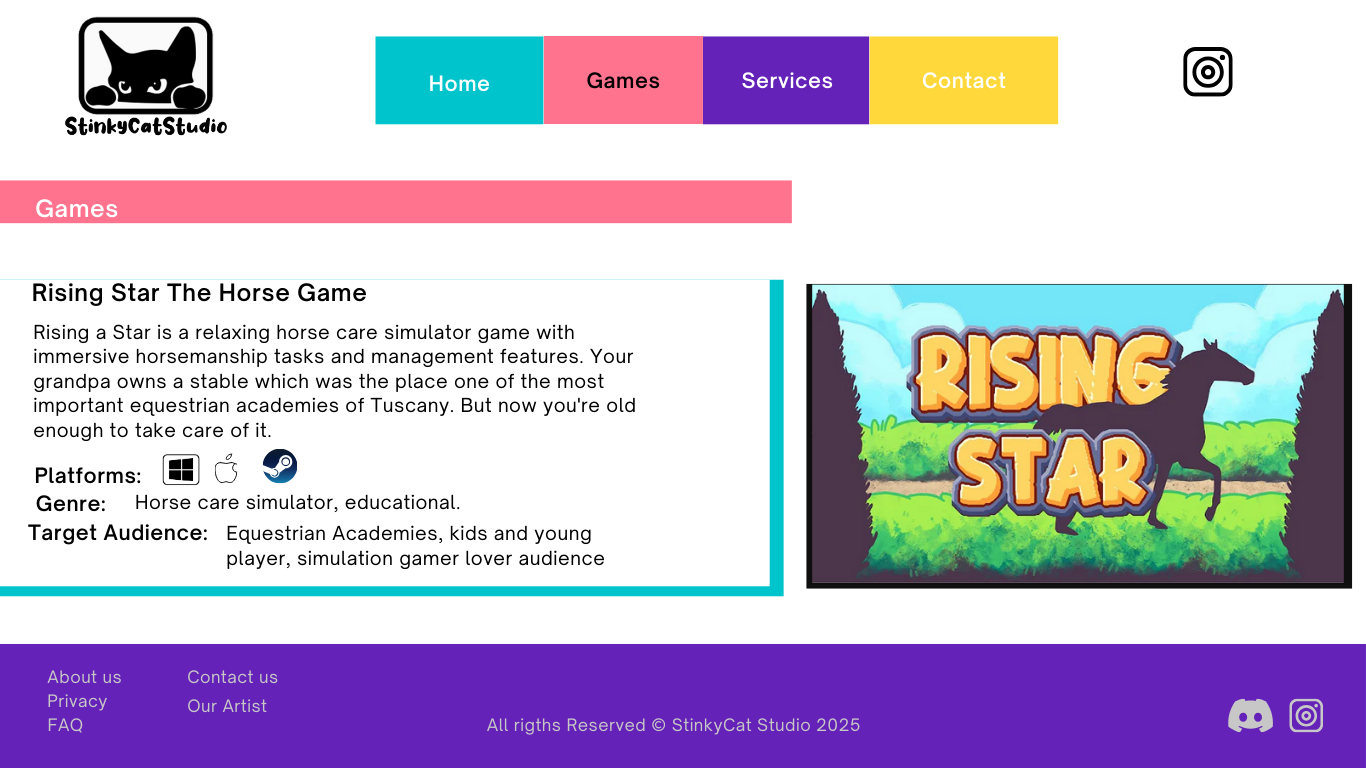
The proposed user interface is based on a vibrant color scheme designed to appeal to young audiences or children's products companies/ schools. For this reason, a six-color palette featuring bright, energetic tones has been selected, with each page distinguished by its own unique color. Additionally, to enhance the playful and friendly aesthetic, the rounded font "DynaPuff" (from Google Fonts) has been chosen. For seamless navigation, a sticky navigation bar will be implemented, allowing users to easily switch between pages at any time*.*

### Color Palette

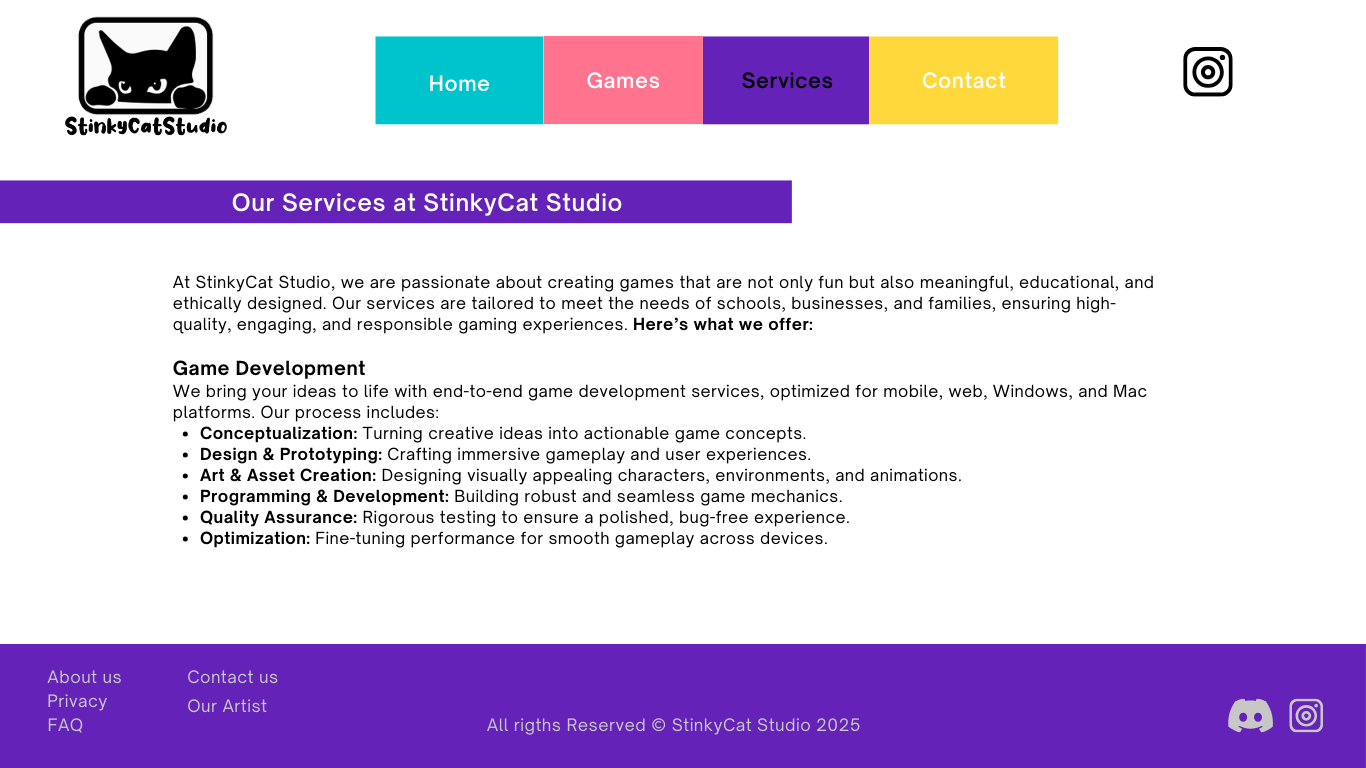


### Home Page

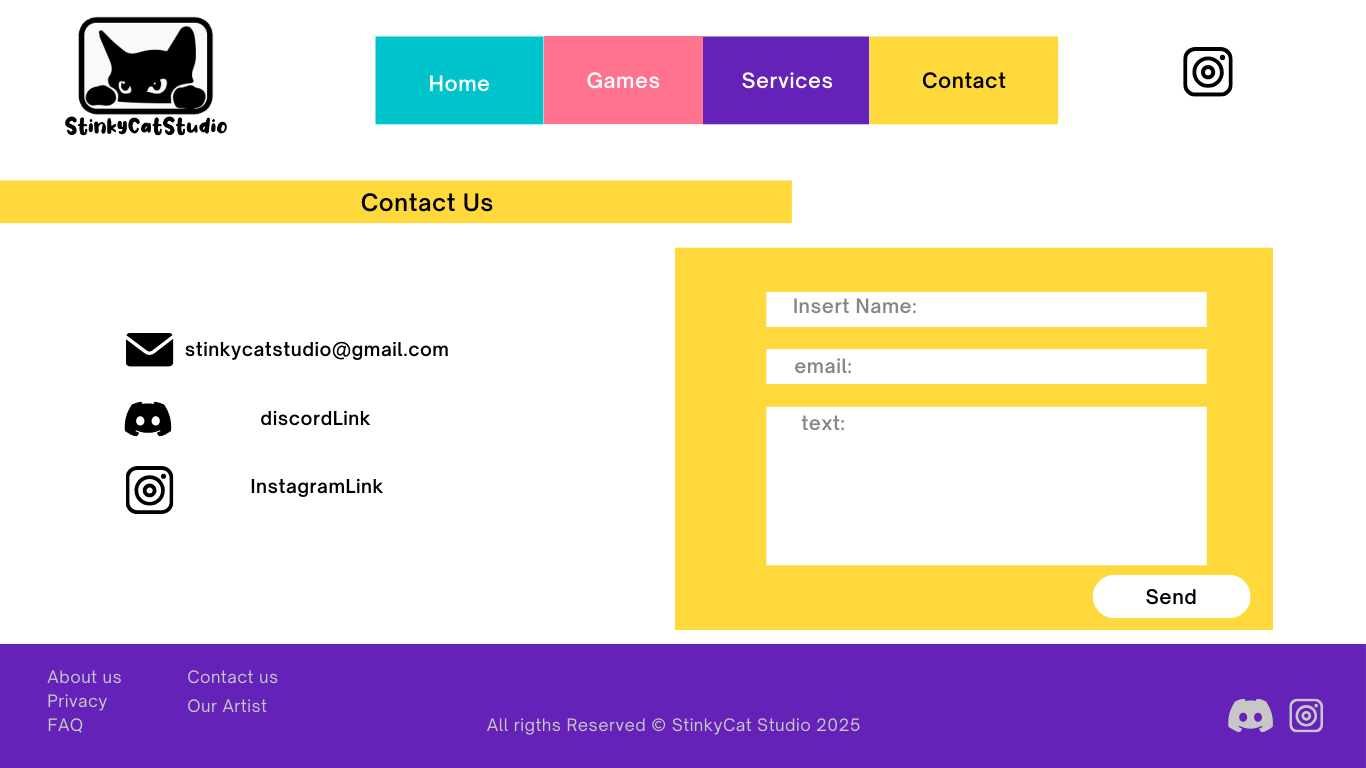
### Games Page



### Services Page



### Contact Page

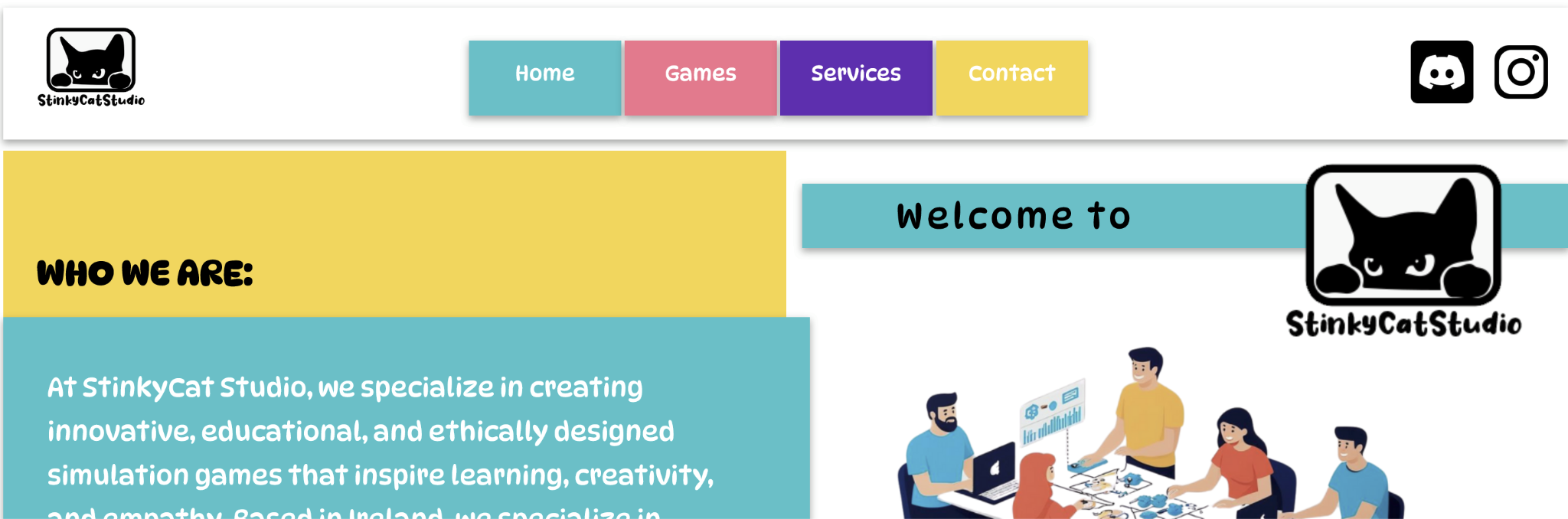


# Upload the Site to a Web Server

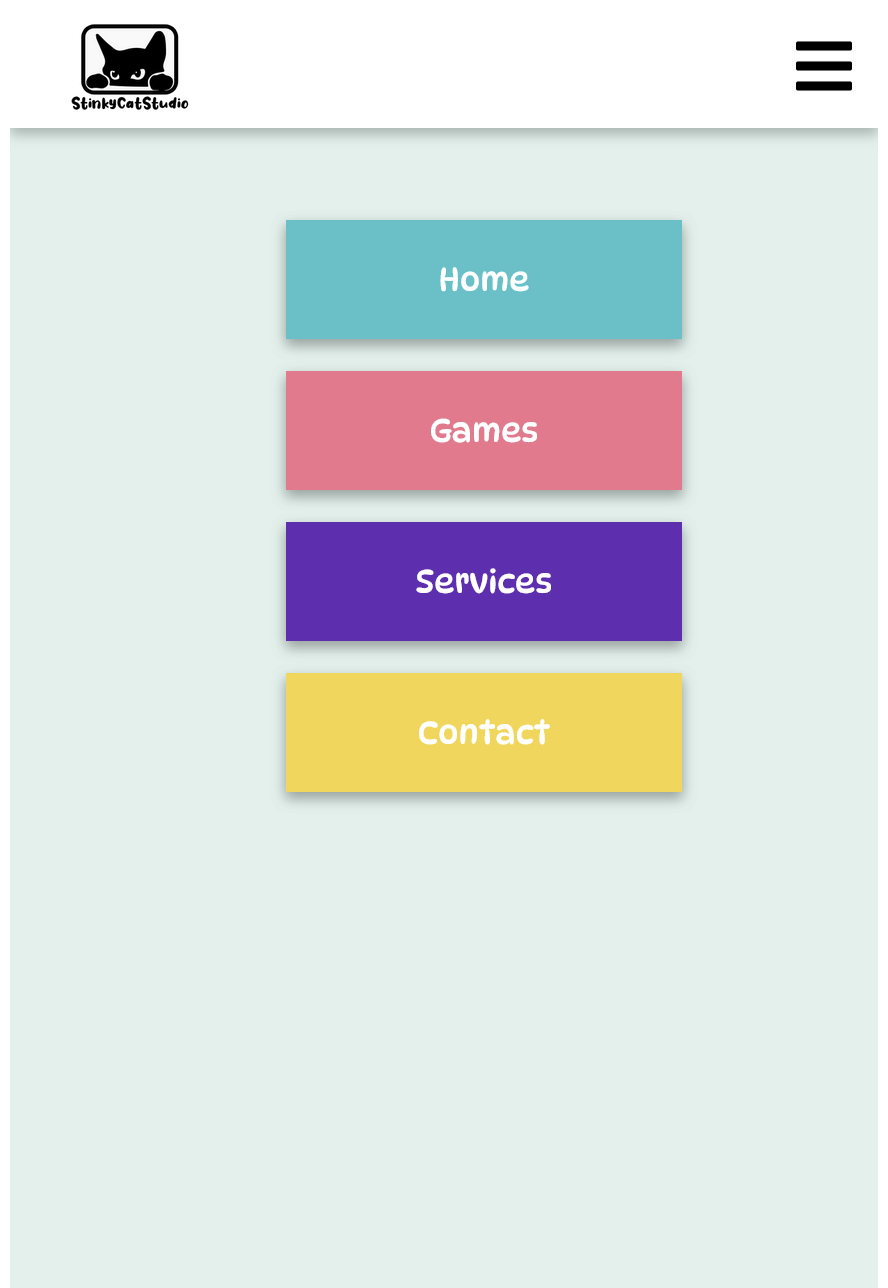
To upload the website to a web server, you’ll need a VPS (Virtual Private Server) since the site was built with Express and requires Node.js. Start by setting up the VPS, installing Node.js, and then cloning the project from GitHub. Once the files are in place, install all necessary dependencies using npm install. Next, connect the server to your database (such as MongoDB or MySQL) to enable data storage and retrieval. Afterward, secure the website by obtaining an SSL certificate (e.g., via Let’s Encrypt) to enable HTTPS encryption. Finally, link your server to a web hosting service using the server’s assigned IP address to make the site publicly accessible.

# Test WebSite Functionality

### Navigation bar

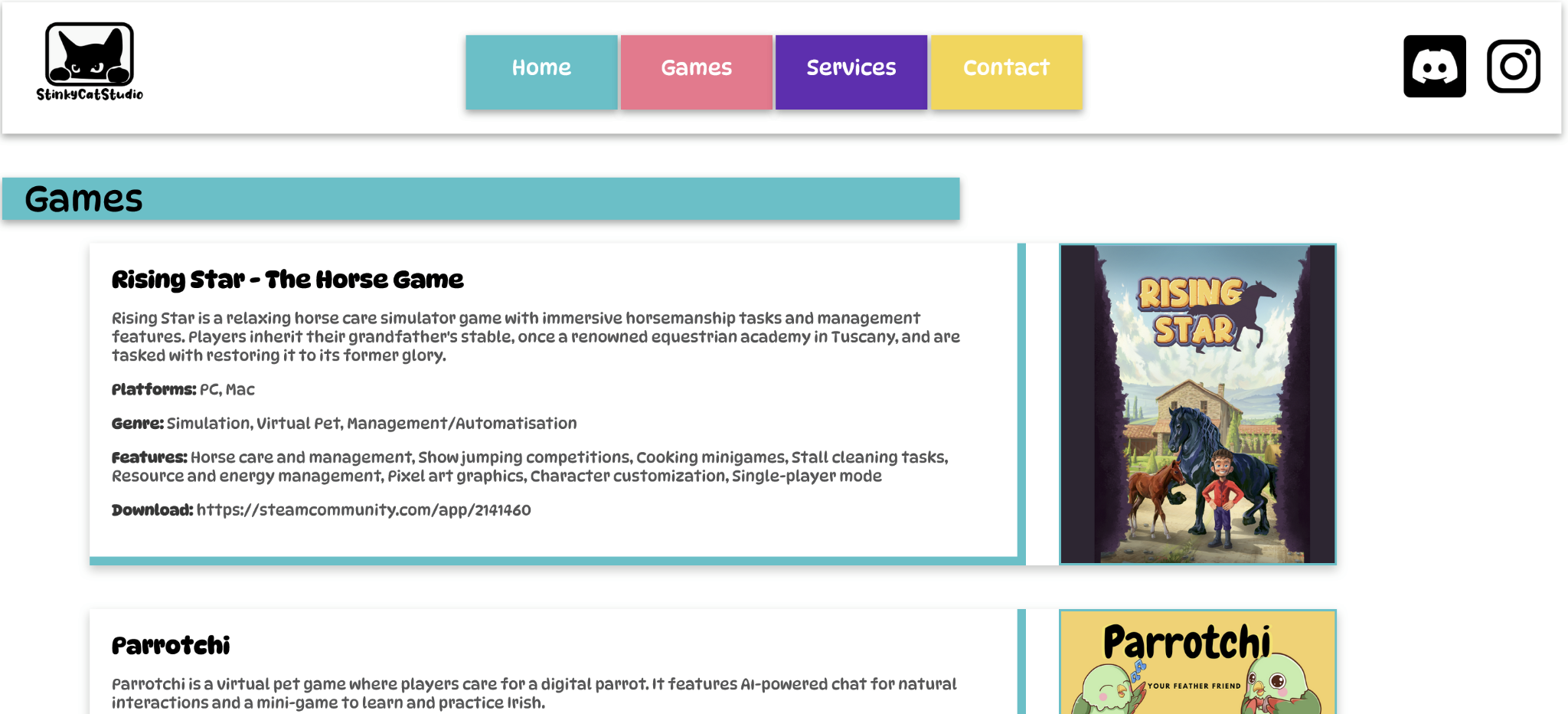


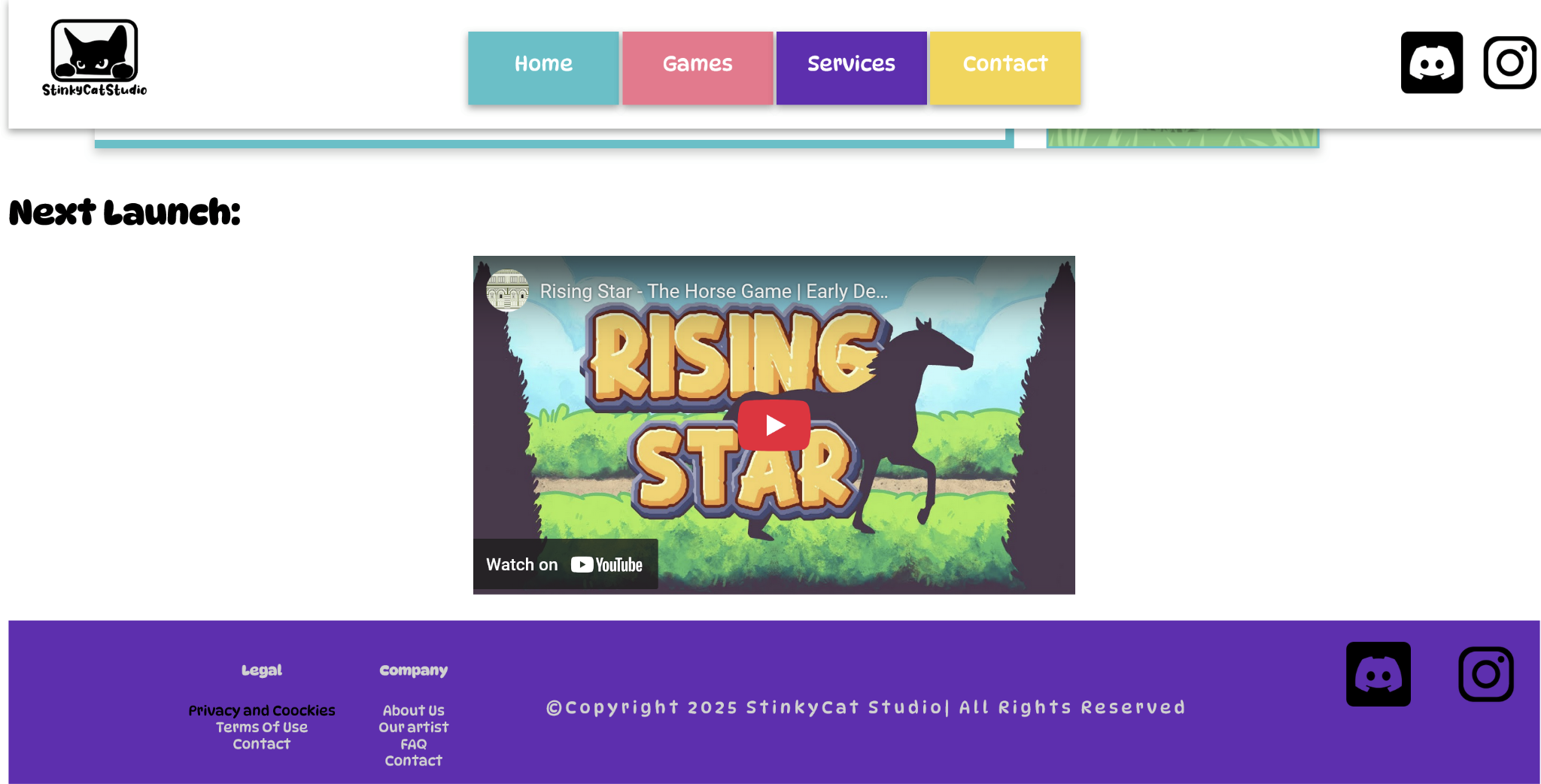
The navigation bar enables users to switch between pages at any time. It remains fixed (sticky) at the top of the screen and highlights the current page with differently colored text. Since the entire website is responsive, the navigation bar adapts to mobile screens by transforming into a collapsible menu, which can be expanded or collapsed using a hamburger icon.



### Games Showcase

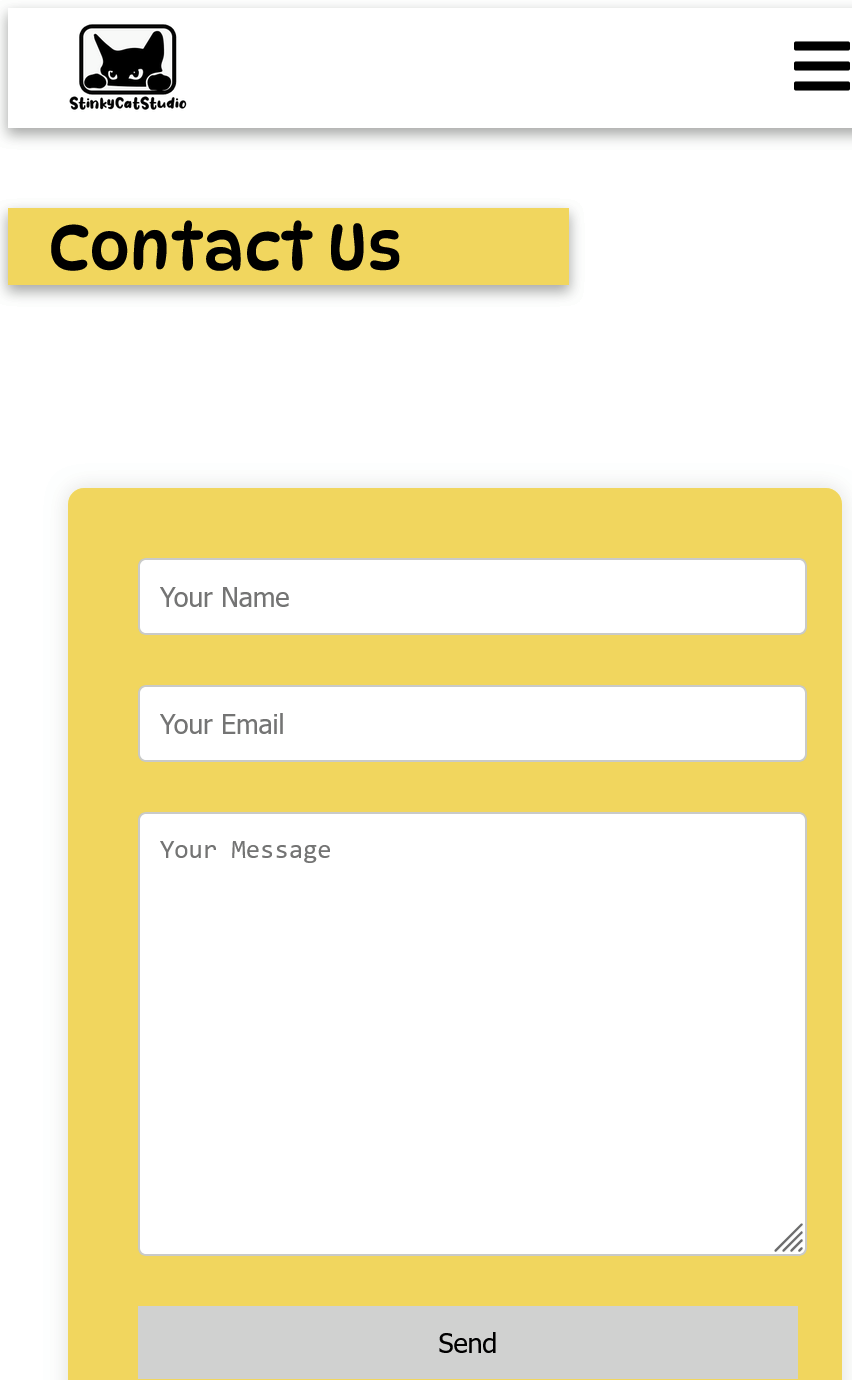
The game page displays all the company's available games, featuring an image, basic details, and an access link for each title. The game information is dynamically retrieved from a JSON file. At the bottom of the page, users can watch an embedded YouTube video showcasing the trailer of one of the featured games.





### Contact Form

### 

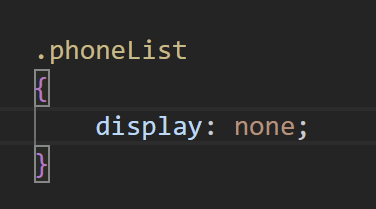


The website features a contact form that allows users to reach out to the company. Submitted information is securely stored in a database. The form is fully responsive, ensuring seamless functionality across all devices, including mobile screens.

# Issues Encountered

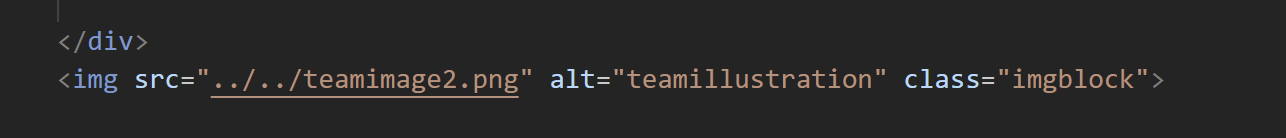
### 1.Mobile Side Menu

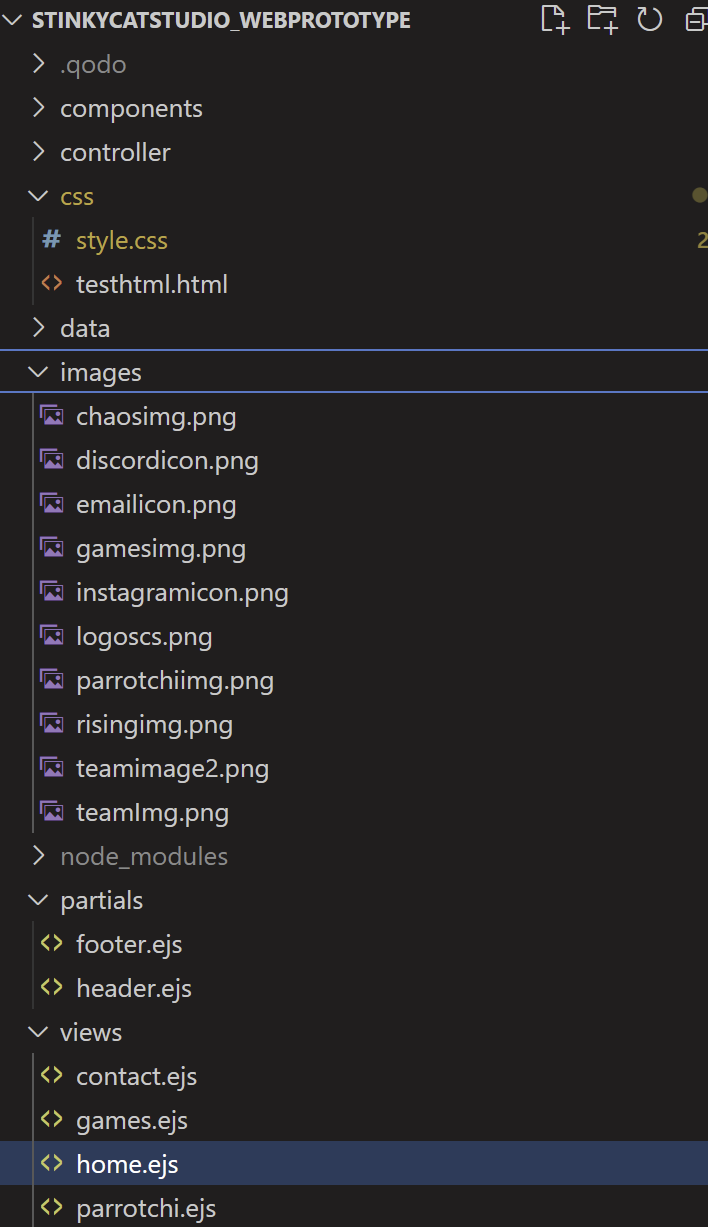
The mobile side menu does not close after being opened when the user resizes the screen.

**Solution:** Use the correct ID and disable the links in the style.css

### 2.Images issue

Images are not being uploaded properly.

Solution: with node js the path for the images have to be different, the img have to be addresses to the root folder 



# Future Planning

### Updating Text:

To update text content in the project, open the folder in **Visual Studio Code**. If you need to change the font, you can find it in the **header partial**, where a **Google Font** is currently being used. For modifying text styles—such as size, color, or spacing—locate the corresponding **class** or **ID** in the style.css file. This ensures consistent styling across the application while allowing for easy customization.

### Creating New Pages

To create a new page, you must first create an **.ejs file** in the **views folder**. Once the file is created, the next step is to add the corresponding route in the **route controller file** to ensure the page is accessible. Additionally, all new pages should maintain consistency by following the existing **layout structure** and **color palette** for a cohesive user experience. For example:

router.get('/newPage', function(req, res){

var title = 'new page title';

res.render('newPage', {title: title})

});

### Inserting Images

To properly insert an image on the website, first place the image file in the project's images folder. When referencing the image in your code, use a root-relative path to ensure it loads correctly. It's crucial to use high-quality images that are properly optimized for web display, balancing quality with file size for optimal performance. For example, you would include an image using HTML like this: <img src="/images/sample.jpg" alt="descriptive text">. Remember to always include appropriate alt text for accessibility and consider using modern formats like WebP for better compression. The image should be sized appropriately for its intended display dimensions while maintaining sharpness on all devices.

<img src="../../teamimage2.png" alt="teamillustration" class="imgblock">

### Creating Links

**External links**: use anchor element in this way

<a href="https://discord.com/"></a>

**Internal links:**

<li class="navLink yellowC"><a href="/contact">Contact</a></li>

### Uploading Updated Content

Before uploading updated content or tools, ensure all modules are properly updated on the server. Verify that everything functions correctly by conducting thorough testing across different browsers. Additionally, confirm that all versions (dependencies, libraries, etc.) match those on the server to prevent compatibility issues. This process helps maintain stability and ensures a seamless experience for users.