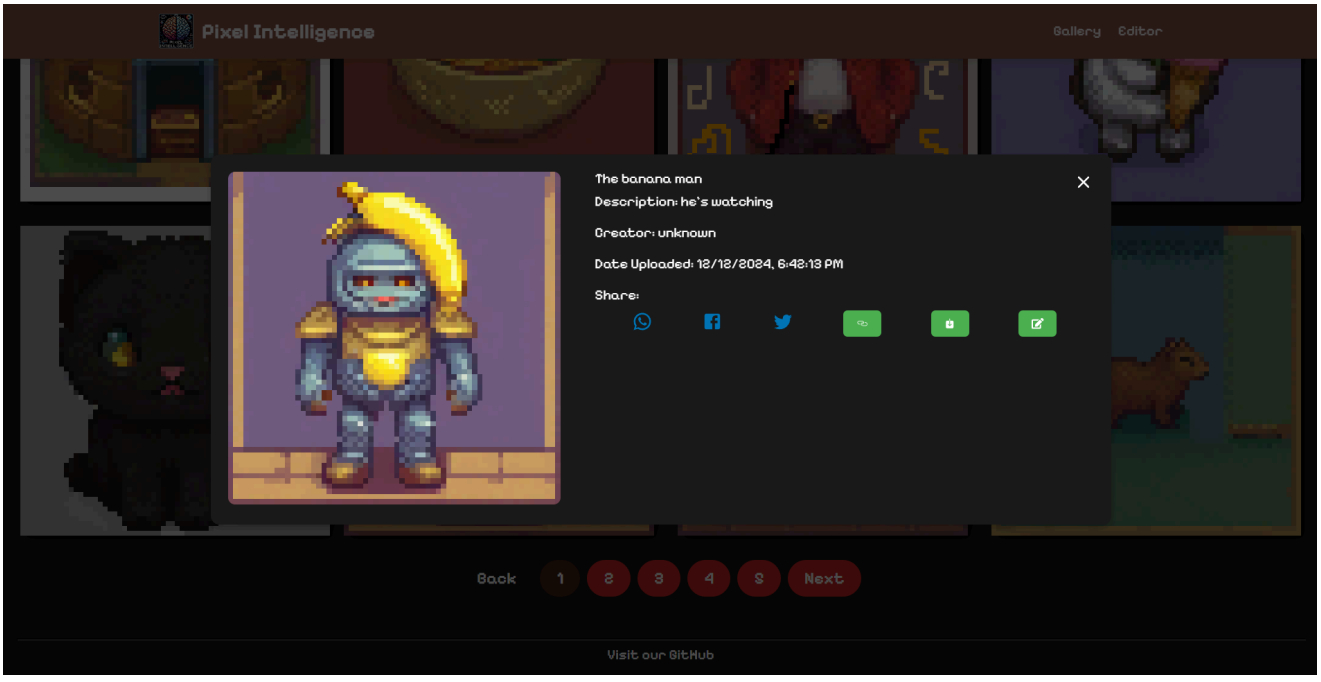
Group 21, Pixel Intelligence Part 4 Reflections 

A Pixel Art Editor with generative AI functionality, image downscale processing, image export, and a showcase gallery. Comp229 Group Project (Fall 2024).

> *Latest live demo can be viewed on Netlify*

Group Members:

- Connor "ConnorBP" Postma (301005412)
- Yoon "superYM222" Min (301317593)
- Sanjeevkumar "sanjeev-cs" Chauhan (301480021)
- Parmila "Parmilashams" Shams (301426195)



Steps to use app

open https://229-pixelai.netlify.app in your desktop or mobile browser, feel free to browse the gallery listing pages and click on any image to view it. Then you may download, share, or even edit that photo with the provided options on the popup. If you click edit it will bring you to the editor and ask if you want to overwrite the current canvas. You can also get to the editor from the top bar. Once in the editor you will see all your document options in the top buttons or context menu, and all

drawing tools on the left. In the top right there is also a shortcut to the gallery on desktop mode, or on mobile it is in the context menu. You may now draw with your cursor or finger, or click on the + icon to generate a new ai image to start with. Once you have an image you like you can download it in json form or in png form with the top buttons, then you can import it again later. On the top right of the buttons is also a share option to share your image on the front page gallery.

For a full list of completed tasks please see our issues tracker on Github

Connor "ConnorBP" Postma (301005412)

This project was sort of intense, but it was very fun. I am very tired and have seen many of javascript's types of bugs at this point. There's still more stuff that can be extended on the program to make it even cooler, but I am quite happy with the feature set and code quality that we got it to for presentation day. Pixel art image scaling works flawlessly, the ai generator makes really cool stuff and is fairly responsive (although the progress bar's prediction code needs some adjustment), the whole data backend in the webapp is page reload resistant in its state management so that users don't loose their in progress drawings or image generation requests, the gallery pagination and gallery viewer and sharing works great, the image caching system is fast and accurate, and the existing drawing features are robust. We didn't end up adding a lot of the fun stretch goals to this because instead the focus was on making what was already there as solid as possible such as the fill bucket algorithm, drawing interpolation, the reload resistant image generation lifecycle (state machine of idle, sending request, waiting, failed, done, etc), making the components more composable (such as the ConfirmationPopup), implementing custom hooks, and many other things to make this app run great and be a good basis for further extensions. Over all I am very happy with the result, and hope to keep this project alive for the foreseeable future as it is just so neat. I would like to specially thank Sanjeev Chauhan for diving in head first into some of the more complicated tasks like the image generation and pagination api backend. I had some big tasks on my plate already, so the fact that he was willing to get his hands dirty on some big tasks really helped make this project possible. I would also like to thank Parmila Shams for their popup component designs and Yoon Min for wrangling the extended editor tooling. This project has helped me a lot with my collaboration skills i think, and it sort of brought me back to when I was new to coding and the sorts of things I didn't really know yet. Guiding my team through technical issues and showing different ways of approaching a problem or code design has been a learning experience for sure. I established fairly rigid standards in code review during this project, which may have made some of the tasks take a little longer, but I hope we have all come away from this a little stronger in our coding skills. :) Thanks for your hard work everyone!

Sanjeevkumar "sanjeev-cs" Chauhan (301480021)

When I started to work on this project I did not much about the React and how the backend works behind the UI. After spending so much time doing this project it help me a lot to learn so many things about React and how the components in react works. Additionally, when I started to work on the image generation api endpoints it seem really hard in the beginning but after discussing with my team member Connor his explanation really helped me to think through the solution to how to make the api work. Finally, just before an hour before our group presentation I found a small CSS error I can say and I also implemented the image creation cancellation button but We forgot to merge the branch to the main thanks to the a person, he will probably know I'm talking about him if he reads this 😂. Anyways, it was quite fun to work on this project and with the contribution and collaboration of our group members we were able to finish thi. So Kudos to every group members 🎉 🎉 🍾. We do have many stretch goals for this application and will work on that also but for now, at least for today, we are done 😁. Lol I wrote quite a bit without noticing. XD

Yoon Min(301317593) - Github: superYM222

This was my first time participating in a project like this. This project taught me how to share code with others using GitHub. In addition, if Conner finds bugs, he uploaded to the GitHub issue page, we can assign them to ourselves to ensure without duplication. At first, GitHub seemed very challenging, but I learned a lot by making many mistakes. I learned about React, and working with multiple pages simultaneously was difficult. However, by looking at other people's code, I realized there are many different ways to implement things. Also, I worked on fixing bugs and improving editors in the features of the left toolbar. one of the issues was with the eyedropper tool—it worked on Chrome but wasn't supported in Safari or mobile. To address this, I added an HTML color picker feature. I also tried to add a gridLine on/off button, but Connor fixed my code, and I learned how to simplify it and make it more reusable from his code. Additionally, I attempted functionality to close a window when clicking the mouse on the background or pressing the ESC key. However, I faced a warning and couldn't resolve it. The issue seems to be related to breaking a React rule.

Parmila "Parmilashams" Shams (301426195)

This project was the most challenging project I have had. It was also an amazing learning experience. In this project I learned how to code with different components. I learned is from challenging I faced and then ask from chatGpt and Conner to solve it.

I sometimes forgot where I had written specific CSS styles because different components required different stylesheets. When I tried to adjust the gallery layout, I accidentally overwrote the Navbar styles, which broke the design on mobile devices. It took hours of debugging and learning how to use better naming conventions for

classes and organizing CSS files to fix this issue. This taught me the importance of keeping my styles modular and organized.

Another challenge was losing my work multiple times. For instance, I worked for two days on styling and functionality changes, thinking I had successfully pushed my code to GitHub, only to find out later that I hadn't created a pull request. My teammate helped me retrieve the code, but I learned to double-check my commits and ensure I followed the correct GitHub workflow.

Totally I can say I got valuable skills in GitHub collaboration, React.js, and CSS styling. I learned to debug more efficiently, organize my code better, and work with dynamic layouts. This project taught me the importance of teamwork, and I am especially thankful to my team members, particularly Connor, for their support and guidance throughout the process.

Requirements

- ✅ have a maximum canvas size of 64px by 64px
- ✅ have a minimum canvas size of 8px by 8px
- ✅ allow users to select a resolution for their pixel art image on start
- ✅ load with a sensible default canvas size initially
- ✅ allow a prompt to generate an image on canvas creation
- ✅ store all editor state (especially pixels) on page reload using local storage
- ✅ allow the user to download a png copy of the image
- ✅ have drawing tools including: pencil, eraser, clear, eyedropper, and fill-bucket
- ✅ have a color selection tool with primary color, secondary color, and a swap button
- ✅ have a database for storing gallery entries
- ✅ have some sort of storage for storing gallery uploads (maybe s3 bucket)
- ✅ allow users to share their creations

Stretch Goals

- ◯ Crochet pattern export
- ◯ encoded URL share codes for pixel art (could use RLE encoding format by ConnorBP)
- ◯ extended color palette presets
- ◯ additional file export formats (popular sprite editor formats, JSON, WebP, etc)
  - ✅ JSON

- ◯ login page (user login system)
  - ✅ JWT sessions
- ◯ transaction support / undo-redo
  - ✅ partial or initial implementation started for line and bucket drawing

Extra stuff we did

- ✅ built in support for first class light and dark modes using the modern prefers-color-scheme media query. My eyes are saved!
- ✅ Full mobile phone support for drawing and everything
- ✅ More detailed, and elaborate, stable diffusion generation system using style Loras, prompt injection, and stable diffusion
- ✅ share to editor from gallery page
- ✅ extended gallery listing metadata (name, description, author, date)
- ✅ adaptive grid lines
- ✅ fully browser refresh resistant state management system
- ✅ composable pop ups
- ✅ responsive design
- ✅ advanced modern css styling such as variables, calculation, animation, and filters
- ✅ nearest neighbour scaling mode for upscale
- ✅ dynamic canvas scaling tool
- ✅ adaptive color picker
- ✅ image import mode (turn any image into pixel art)
- ✅ netlify serverless functions based deployment for express backend
- ✅ input validation on api endpoints
- ✅ probably so much more