



ART  
GUARDIAN



Team Blue

CS411 Fall 2022 Prototype Presentation #2



# Table of Contents

3. Team Roles
4. Sprint 0 & 1 Breakdown
5. Sprint 2 Breakdown
6. Sprint 3 Breakdown
7. Sprint 4 Breakdown
8. Sprint 5 Breakdown
9. Sprint 6 Breakdown
10. Features Roadmap
11. Team Roles and Responsibilities
12. Accountability Synopsis
13. Features Completed
14. Features in progress
15. Prototype Changes
16. Near Term Priorities
17. Concerns

# Team Blue



Carlo Diaz  
Frontend/Webmaster



Paolo Ihde  
Frontend/Backend



Brandon Kidd  
Database Specialist/Backend



Michael Thompson  
Backend/Frontend



Autumn Roberts  
Team Leader/Frontend



Tobin Zheng  
Backend / Web Master



Spencer Hite  
Database Specialist/Backend



# Sprint 0 & 1 Task Breakdown

Team:

- Setup Dev Environments
- Setup Cloud Services and Database



# Sprint 2 Task Breakdown

## Front-End:

- Create Landing Pages
- Create Sign in and login pages

## Back-End:

- Tested an Image Matching algorithm

## Database:

- Finish GraphQL data table for Settings.
- Start GraphQL data table to hold a images' metadata.
- Generate common GraphQL mutations and queries to read/write to DynamoDB database.
- Launch AWS S3 bucket to hold User Artwork images.



# Sprint 3 Task Breakdown

## Front-End:

- Create user art upload and gallery
- Create skeleton code for DMCA Generation
- User Account Security and Encryption
- Login Cookies

## Back-End:

- Create whitelisting functionality
- Implement Image Matching as part of back-end
- Implement detection of Art on NFT Marketplaces
- Create fake NFTs

## Database:

- Finish GraphQL data table to hold a images' metadata.
- Start GraphQL data table to hold DMCA copyright information.
- Update GraphQL queries and mutations to include any new or modified data tables.



# Sprint 4 Task Breakdown

## Front-End:

- Execute DMCA Takedown Requests
- Account Settings
- Account Profile

## Back-End:

- DMCA takedown generation
- Integrate and test Gmail API

## Database:

- Finish GraphQL data table to hold DMCA copyright information
- Start GraphQL data table to hold notification metadata
- Update GraphQL queries and mutations to include any new or modified data tables.



# Sprint 5 Task Breakdown

## Front-End:

- Art Protection Guidelines
- App Tutorial
- Generate Push Notification

## Back-End:

- DMCA Cataloging
- DMCA Tracking

## Database:

- Finish GraphQL data table to hold Notification metadata
- Database maintenance



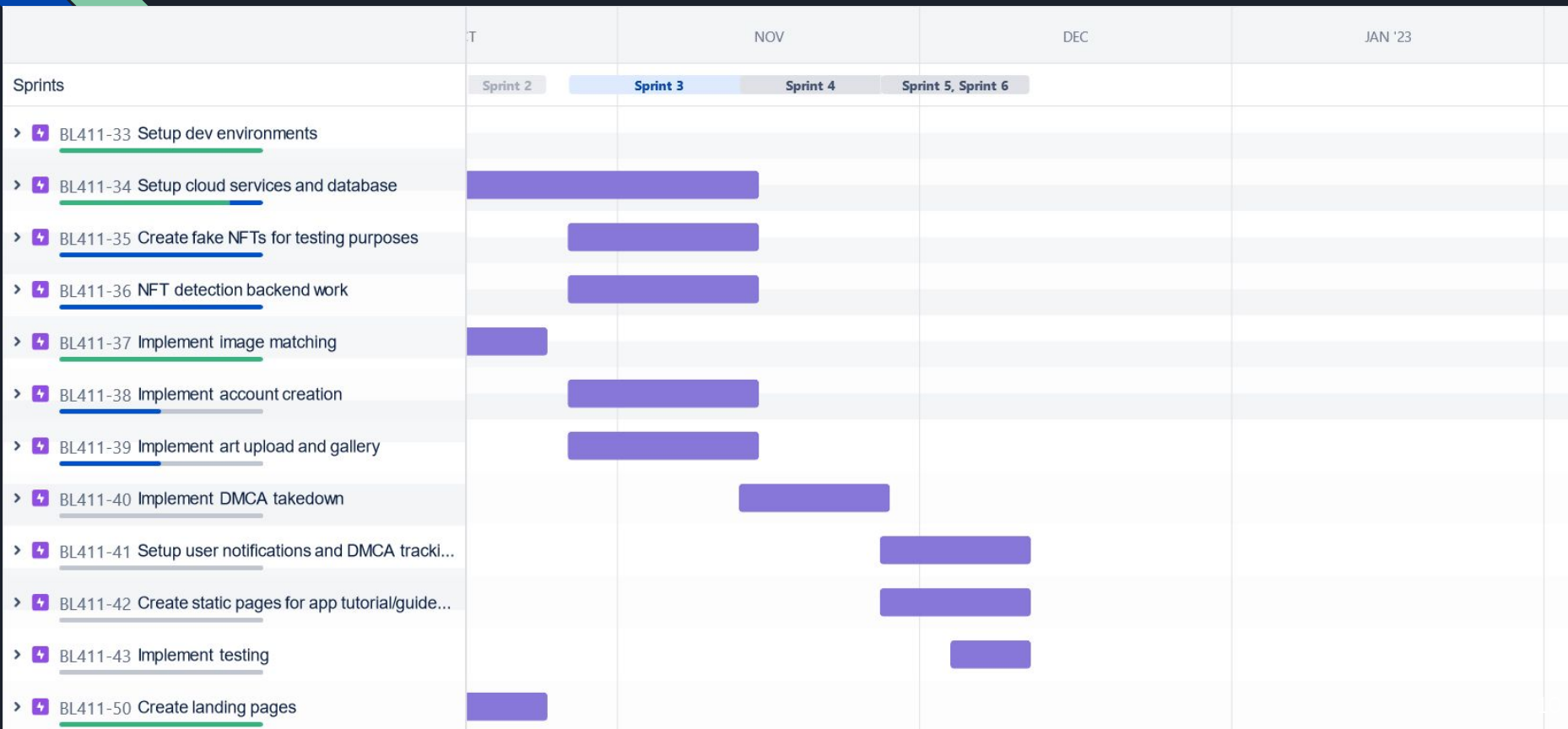


# Sprint 6 Task Breakdown

Team:

- Create Testing Environment
- Prototype Test
- Project Cleanup

# Features Roadmap





# Team Roles and Responsibilities

- Scrum Masters - Autumn, Tobin
  - Sprint planning
  - Alternating leaders for team meetings (Scrum, Stand-ups, etc.)
- Frontend Developer - Carlo, Autumn, Paolo
  - AWS Amplify, ReactJS, CSS development
  - App Feel and overarching themes
  - User Accessibility
- Backend Developer - Michael, Tobin
  - Lambda, Step Functions, Python, OpenCV
  - Implementation of algorithms:
    - Whitelisting
    - Counterfeit art detection
    - Image matching
    - DMCA
- Database Developer - Brandon Kidd, Spencer Hite
  - AWS Amplify Database implementation and maintenance.
  - GraphQL mutations/queries implementation and maintenance.



# Accountability Synopsis

- Keep track of commits and pushes
- Assigning members to tasks on Jira
- Honor System:
  - I pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community it is my responsibility to turn in all suspected violations of the Honor Code. I will report to a hearing if summoned.
- Weekly meetings to discuss progress

# Features Completed

▼	🔧	BL411-33	Setup dev environments		
	📁	BL411-28	Setup version control (git)	DONE	
	📁	BL411-27	Setup dev environment	DONE	
▼	🔧	BL411-34	Setup cloud services and database		
	📁	BL411-30	Startup AWS Amplify and Database	DONE	BK
	📁	BL411-29	Configure AWS, Create Team Account	DONE	BK
	📁	BL411-45	Add Data Models (user/profile information) to Amplify Backend Environment	DONE	BK
	📁	BL411-16	Access Database/S3 for image reading	DONE	SH
	📁	BL411-22	Access Database for user information reading	DONE	BK
▼	🔧	BL411-37	Implement image matching		
	📁	BL411-19	Invoke image matcher to test matching values	DONE	MT
▼	🔧	BL411-50	Create landing pages		
	📁	BL411-49	Create skeleton code for sign-up page	DONE	AR
	📁	BL411-48	Create skeleton code for sign-in page	DONE	PI
	📁	BL411-51	Create skeleton code for landing page	DONE	CD

# Features in Progress

▼	⚡ BL411-34 Setup cloud services and database		
	📁 BL411-52 Add Art/DMCA data models and complete the Database Schema	IN PROGRESS	BLK
▼	⚡ BL411-35 Create fake NFTs for testing purposes		
	📁 BL411-20 Make fake NFT to test whitelisting	IN PROGRESS	MT
	📁 BL411-18 Make fake NFT to test flagging system	IN PROGRESS	MT
▼	⚡ BL411-36 NFT detection backend work		
	📁 BL411-13 Whitelisting of minted NFT	IN PROGRESS	MT
	📁 BL411-12 Provision of NFT information correlating to user artwork	IN PROGRESS	TZ
	📁 BL411-7 Detection of art on NFT marketplaces	IN PROGRESS	TZ
	⚡ BL411-37 Implement image matching		
▼	⚡ BL411-38 Implement account creation		
	📁 BL411-1 Account sign-in to access app	IN PROGRESS	PI
	📁 BL411-11 User account security and encryption	IN PROGRESS	SH
▼	⚡ BL411-39 Implement art upload and gallery		
	📁 BL411-5 Art upload	IN PROGRESS	AR



# Prototype Changes

- Upload and Gallery now one page
- Created Logo
- Changed gallery database from Dynamo to S3
- Utilized amplify features
- Major changes to Art Detection feature:
  - Images will not be batched
  - Future of periodic searches unclear
- Adjusted weights for Image Matching Algorithm



# Near-term Priorities

- Login Cookies
- Integrate algorithms into workflow / step functions
- Finish database implementation





# Concerns

- Art Gallery separating and showing art based on user
- Time constraints
- Team Availability
- Difficulty curve
  - Step Functions / AWS Lambda