American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

DIGITAL CANVAS HUB

A Software Quality and Testing Project Submitted
By

Semester: Spring_24_25				Section:	Group No:
SL	**SN**	**Student Name**	**Student ID**	Individual Contribution (in %)	Total Marks: 50 Earned Marks:
A		Md. Asheke Rabbi	21-44852-2	100%	
B		Yeameen Aziz Abdullah	21-44886-2	100%	
C		Nabiha Tahsin	21-45685-3	100%	
D		Tanvir Arafat	21-45692-3	100%	

The project will be Evaluated for the following Course Outcomes

EVALUATION CRITERIA	**Total Marks (50)**	
Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions	[10 Marks]	
Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements)	[10 Marks]	
Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases	[10 Marks]	
Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk	[10 Marks]	
Approval, Format, Submission, and Defense	[10 Marks]	

Software Test Plan

for

<Digital Canvas Hub>

Version 9.0

Prepared by <Md Asheke Rabbi, Yeameen Aziz Abdullah, Nabiha Tahsin, Tanvir Arafat>

<American International University-Bangladesh>

<18 June, 2025>

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	2025.06.15	Yeameen Aziz	First Draft
0.2	2025.06.16	Nabiha Tahsin	Second Draft
0.3	2025.06.16	Yeameen Aziz	Third Draft
0.4	2025.06.16	Md Asheke Rabbi	Fourth Draft
0.5	2025.06.17	Md Asheke Rabbi	Fifth Draft
0.6	2025.06.17	Yeameen Aziz	Sixth Draft
0.7	2025.06.17	Tanvir Arafat	Seventh Draft
0.8	2025.06.18	Tanvir Arafat	Eighth Draft
0.9	2025.06.18	Nabiha Tahsin	Ninth Draft

1. TEST PLAN IDENTIFIER: DCH-TP-09

2. REFERENCE MATERIALS

[1] M. M. H., "Software Test Plan Manual," Dept. of Computer Science, FST, American International University-Bangladesh (AIUB), 2024.

[2] shashankvenkatesh4, "Test-cases.docx," Course Hero. [Online]. Available: https://www.coursehero.com/file/156655437/Test-casesdocx/. [Accessed: Jun. 19, 2025].

[3] B. W. Boehm, *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[4] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed. New York, NY: McGraw-Hill, 2010.

3. INTRODUCTION

3.1 Background to the Problem

In this new era of digitalization, Artworks are trending to be considered as a strong medium for expressing digital creativity, technological skill and own perspectives through art. But with growing trends, many artists face struggles in gaining proper recognition, enough income through their art and effort in creating them.

The lack of a convenient and artist friendly marketplace or community that can offer the supportive needs of digital artists and curious audiences leads to that problem. Digital artists do not have many opportunities for showcasing their great work of talent and earning through it as traditional ways of arts still hold much bigger priority to society. The increasing popularity of accepting and embracing digital art expression calls for more support, adaptability feature, flexible and technologically supervised platforms, making people drawn to digital art day by day.

3.2 Solution to the Problem

Addressing these challenges, the development of a dedicated digital art marketplace is important. A platform like “Digital Canvas Hub” provides artists with a modern, accessible, and global atmosphere to showcase their portfolios and sell their art securely. Such a system not only supports the visibility and creativity of artists but also encourages innovation by providing tools for interaction, discovery, and transaction for which all in one unified platform.

Although several platforms such as OpenSea, Pinterest, Instagram, and DeviantArt offer alternative solutions, they often fall short in delivering an experience that is specifically crafted to the needs of digital artists and their audiences. “Digital Canvas Hub” is designed to focus these gaps by enabling both emerging and established artists to receive exposure, engagement, and financial support in a more structured and artist-friendly ecosystem. Moreover, the platform itself fosters a rich community and society by bringing together artists, collectors, and creators by supporting cultural exchange and the appreciation of diverse artistic styles.

The uniqueness of "Digital Canvas Hub" lies in its focused integration of features that serve the digital art community in a efficient way. The platform includes customizable artist portfolios, integrated blogging tools for creative storytelling, private messaging functionality, and a secure wallet-based transaction system. Administrative features allow curated content and highlighted featured artists, while advanced search and filtering features enhance user experience and artwork visibility. These thoughtful design choices position the platform as an extensive and future-oriented solution for digital creativity, bridging the broad gap between artists and the audiences.

4. REQUEIREMNT SPECIFICATION

4.1 System Features

1. Registration
<u>Functional Requirements</u>

1.1 The site shall allow users to register as a "User" or as an "Artist"
1.2 The user shall create account with legit email and password on the website
1.3 The user shall choose a unique username
1.4 The email format must be valid
1.5 Password must be minimum 8 digits. Minimum 1 Upper Case, 1 Lower Case, 1 Number and 1 special character such as (! @ # $ % ^ &) character.
1.6 The user shall receive a verification message sent to their supplied email. Users need to click on the link which is given in the email in order to activate account.

Priority Level: High
Precondition: User must have a valid and accessible email address.
Cross reference:

2. System Login
<u>Functional Requirements</u>

1.1 The site shall allow users to register using the given username and password
1.2 The user shall provide the OTP to login
1.3 OTP shall expire after five minutes and can be regenerated at any time.
1.4 The session will end if the OTP verification is unsuccessful three times.

Priority Level: High
Precondition: Valid credentials must be submitted before 2FA
Cross reference:

3. Two-Factor Authentication (2FA)

Functional Requirements

1.1 The system shall send verification code / OTP to the email after they enter their login information.
1.2 If the username and/or pass are incorrectly entered more than 3 times then a CAPTCHA shall be asked the user to solve.
1.3 If the number of login attempts exceeds its limit (5 times), the system shall block the user's account login for one hour
1.4 Users can use "Forgot Password" feature to retrieve their account information through email
1.5 The system shall support optional Two-Factor Authentication (2FA) through an authenticator app.

Priority Level: High
Precondition: User must have a registered and activated account.
Cross reference:

4. Sell Artwork

Functional Requirements

1.1 Artist can upload new artwork by artists along with the title, price, description, and any pertinent tags.
1.2 The system shall allow artists to upload artwork to a maximum file size (e.g., 50MB) and formats (JPG, PNG).
1.3 The system shall provide an option to have a digital watermark added by the system.

Priority Level: High
Precondition: For selling, user must have an "Artist" account.
Cross reference:

5. Art Portfolio/Showcase

Functional Requirements

1.1 The public artworks of each artist shall display in a gallery view in their profile.
1.2 Artists can arrange their works in "Series" or "Collections."
1.3 Without logging in, users can peruse artist portfolios.
1.4 Artists can pin featured artworks to the top of their profile.

Priority Level: Medium
Precondition: Artist must have uploaded artworks.
Cross reference:

6. Purchase Artwork

Functional Requirements

1.1 Customers can use their wallet balance to buy artwork.
1.2 Customers can buy more than one item at once by adding artwork to their shopping cart.

Priority Level: High
Precondition: Artist For purchasing, user must have sufficient currency in their wallet.
Cross reference:

7. Like, Comment & Follow

Functional Requirements

1.1 Users can like and comment on public artworks.
1.2 All users can see the moderated comments.
1.3 Users can receive notifications of new uploads and follow artists.
1.4 Artists can control who can follow or comment on their work.

Priority Level: Medium
Precondition: User must be logged in.
Cross reference:

8. Blog Integration

Functional Requirements

1.1 Artists can post updates, advice, or details about their creative process on their blogs.
1.2 Blogs can include text, images, and embedded video links.
1.3 Users can like and comment on blog posts.
1.4 The administrator can highlight specific blogs on the home page.

Priority Level: Low to Medium
Precondition: Artist account must be verified.
Cross reference:

9. User Profile Management

Functional Requirements

1.1 Users can view their own profiles on the website, complete with artwork, username, and join date.
1.2 With proper verification, users can change their profile details, including their first and last names, bio, and profile picture.
1.3 Artists can modify, remove, or un-list their own works of art.
1.4 Users can view public profiles of other users and artists
1.5 To get updates, users can follow or unfollow artists.

Priority Level: High
Precondition: User must be logged in and verified.
Cross reference:

10. Messaging and Notifications

Functional Requirements

1.1 Users shall able to message each other privately
1.2 Other users can send and receive private messages
1.3 The website shall show notifications for new messages, artwork sales, new followers, and comments on their artwork
1.4 Users can stop receiving messages from other users.

Priority Level: Medium
Precondition: User is logged in and has the username of the recipient.
Cross reference:

11. Search and Discovery

Functional Requirements

1.1 Users shall able to find artwork by searching title, tag, or artist name.
1.2 Users can search for users and artists by their username.
1.3 For search results, the system shall offer sophisticated filtering options (e.g., by price range, color palette, creation date).
1.4 To facilitate discovery, the homepage shall include sections for "Trending Art," "New Arrivals," and "Featured Artists."

Priority Level: High
Precondition: User is logged in.
Cross reference:

12. Transaction and Wallet Management

Functional Requirements

1.1 Users can add money to their wallet through integrated payment gateways (such as PayPal and credit cards) and can use the top-up feature to add funds to their wallet.
1.2 Users can withdraw funds from their wallet to PayPal or a bank account they have linked.
1.3 The system shall maintain complete transaction history, including all top-ups, purchases, sales, and withdrawals, these shall be kept up to date by the system.
1.4 Every financial transaction must be safely recorded and auditable.

Priority Level: High
Precondition: User has a verified account and has linked a valid payment method.
Cross reference:

13. Admin Functions

Functional Requirements

1.1 The administrator can access an admin dashboard, with unique admin credentials.
1.2 "User" or "Artist" accounts can be view, suspend, or delete by the administrator.
1.3 Admin shall be able to approve or disapprove artist verification requests and feature/un-feature artworks on the homepage
1.4 The administrator can access to site-wide data, such as daily revenue, total users, and total sales.
1.5 The administrator can oversee user reports and take the necessary action (such as suspending a user or removing content).

Priority Level: High
Precondition: User must have admin-level credentials and be logged in.
Cross reference:

4.2 System Quality Attributes

QA1 - Reliability: On a monthly basis, the system will be operational 99.5% of the time. Login, search, and transaction success rates for crucial operations shall be 99.9%. The database will be automatically backed up every night by the system.

Priority Level: High
Precondition: N/A
Cross reference: QA3 (Performance)

QA2 - Usability: In an average of five minutes, new users shall be able to register, locate, and buy an artwork. For a user accustomed to using standard e-commerce websites, the user interface will be easy to use and require little to no training.

Priority Level: High
Precondition: User has access to the website.
Cross reference: QA3 (Performance)

QA3 - Performance: Under an average load of 500 concurrent users, all web pages shall load completely in 3 seconds. In less than two seconds, the artwork search function shall produce results.

Priority Level: High
Precondition: N/A
Cross reference: QA1(Reliability), QA6 (Scalability)

QA4 – Flexibility: The system architecture shall be modular, enabling a developer to integrate a new payment gateway in 40 man-hours without compromising already-existing functionality.

Priority Level: Medium
Precondition: N/A
Cross reference: QA5(Testability)

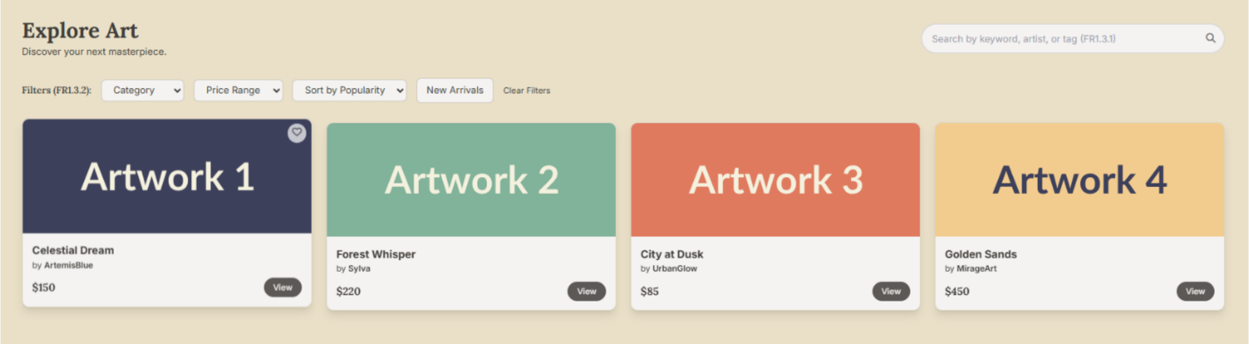
QA5 – Testability: Automated testing shall be made possible by the system. Unit tests will cover at least 80% of the business logic (backend) code. There will be end-to-end automated tests for every significant user workflow.

Priority Level: High
Precondition: Development of a feature is complete.
Cross reference: QA4(Flexibility)

QA6 – Scalability: Over the course of six months, the system architecture must be able to accommodate a 50% increase in user traffic without experiencing any performance issues. In order to manage traffic spikes of up to 1,500 concurrent users, the infrastructure will be built to scale horizontally.

Priority Level: High
Precondition: N/A
Cross reference: All other QAs

4.3 System Interface



Explore Art

Discover your next masterpiece.

Search by keyword, artist, or tag (FR1.3.1)

Filters (FR1.3.2):

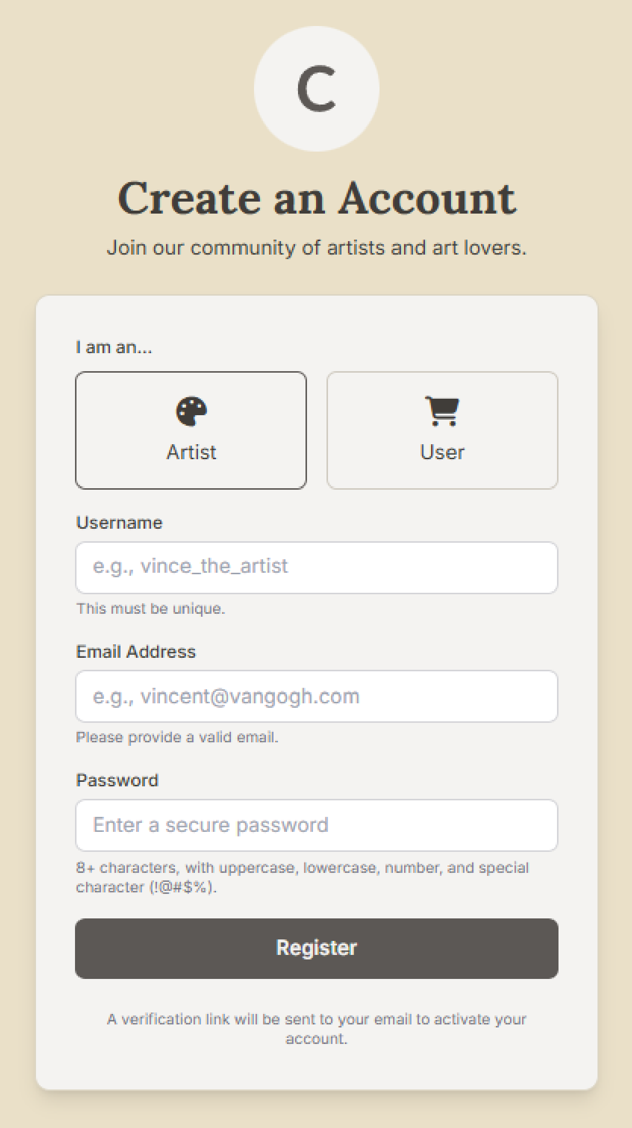
Celestial Dream
by ArtemisBlue
$150

Forest Whisper
by Sylva
$220

City at Dusk
by UrbanGlow
$85

Golden Sands
by MirageArt
$450

Fig.1: Homepage



Create an Account

Join our community of artists and art lovers.

I am an...

Username

e.g., vince_the_artist

This must be unique.

Email Address

e.g., vincent@vangogh.com

Please provide a valid email.

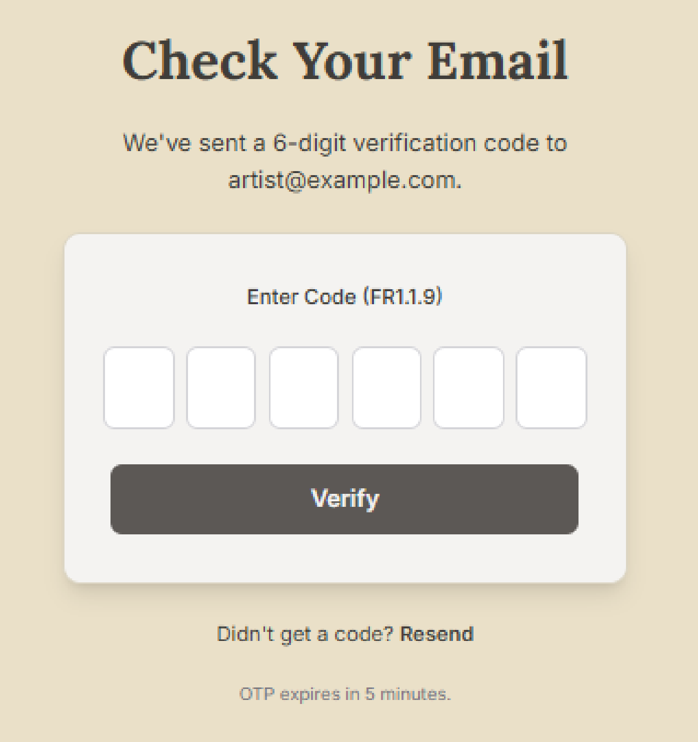
Password

Enter a secure password

8+ characters, with uppercase, lowercase, number, and special character (!@#$%).

A verification link will be sent to your email to activate your account.

Fig.2: Registration



Check Your Email

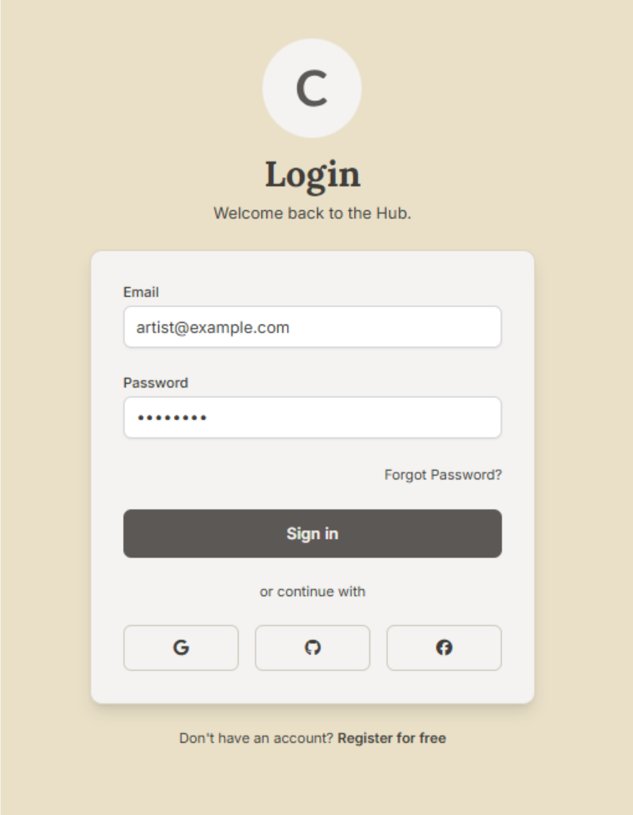
We've sent a 6-digit verification code to artist@example.com.

Enter Code (FR1.1.9)

Didn't get a code? Resend

OTP expires in 5 minutes.

Fig.3: Email Verification



Login

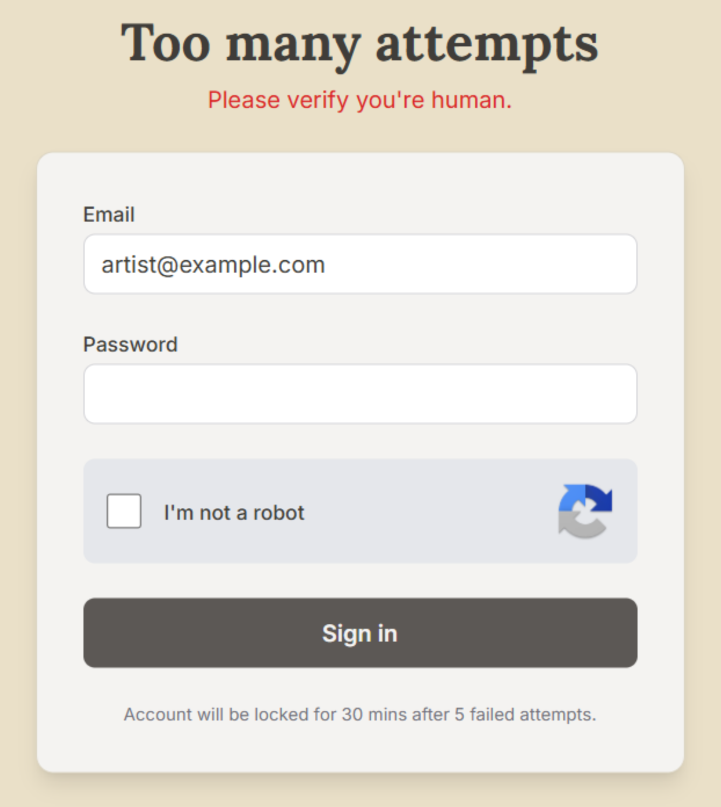
Welcome back to the Hub.

Email: artist@example.com

Password: ••••••••

Don't have an account? Register for free

Fig.4: Login



Too many attempts

Please verify you're human.

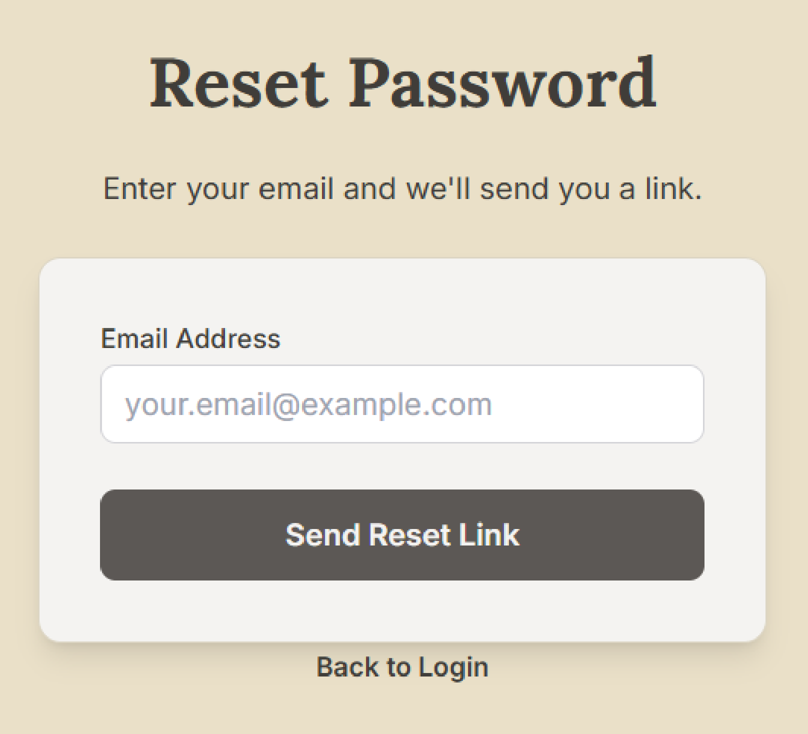
Email: artist@example.com

Password:

□ I'm not a robot

Account will be locked for 30 mins after 5 failed attempts.

Fig.5: Login with CAPTCHA

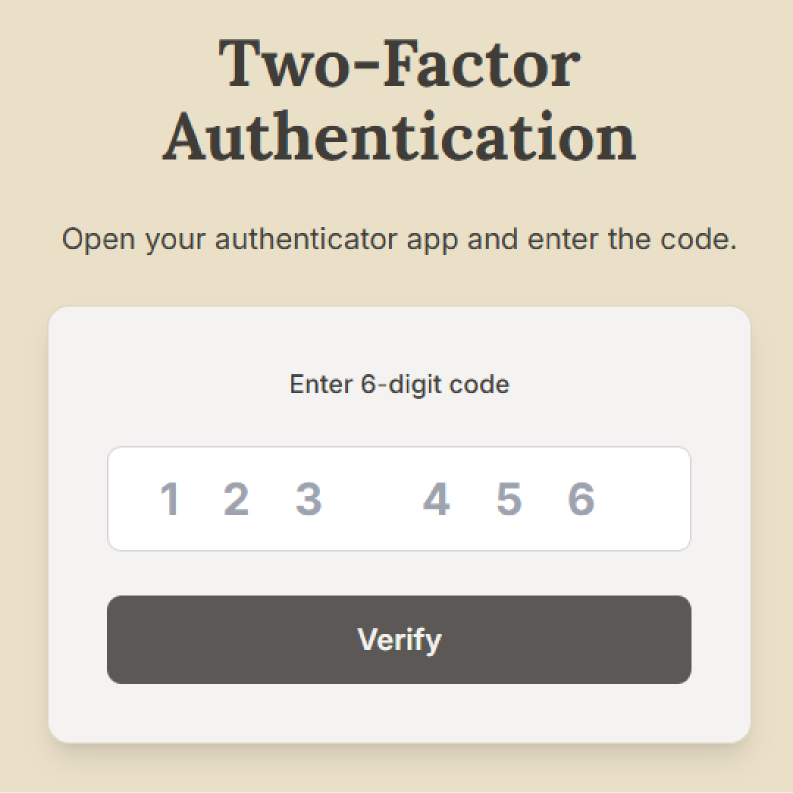


Reset Password

Enter your email and we'll send you a link.

Email Address: your.email@example.com

Fig.6: Reset Password



Two-Factor Authentication

Open your authenticator app and enter the code.

Fig.7: Two Factor Authentication

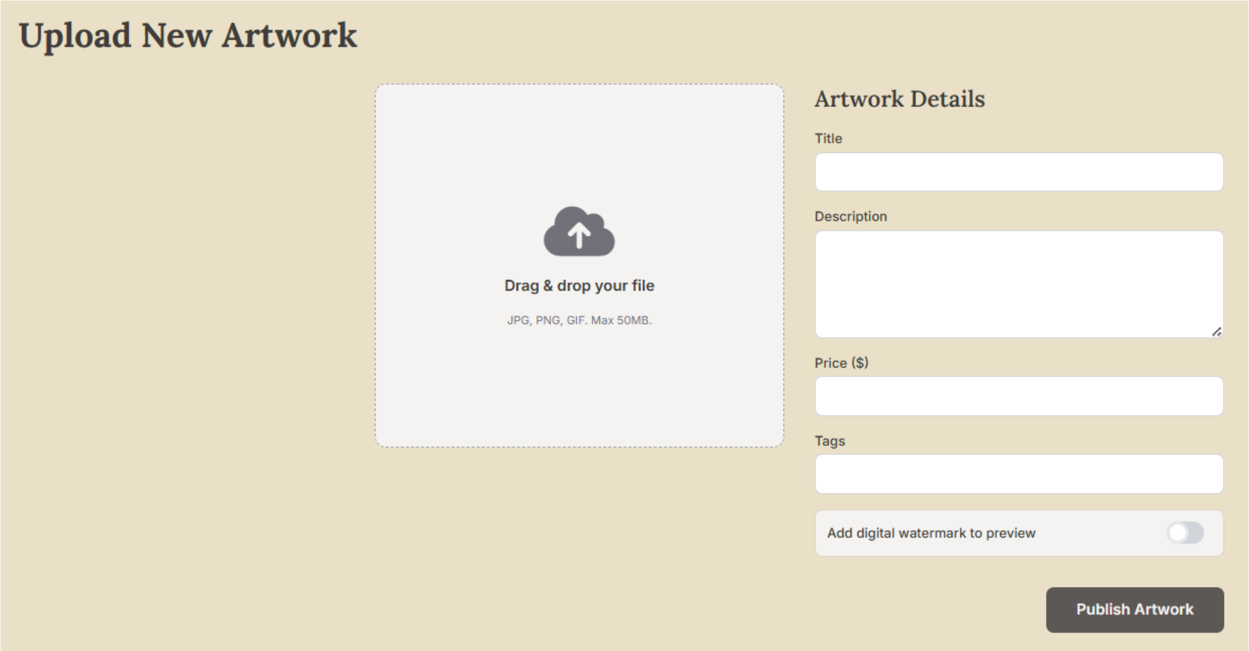
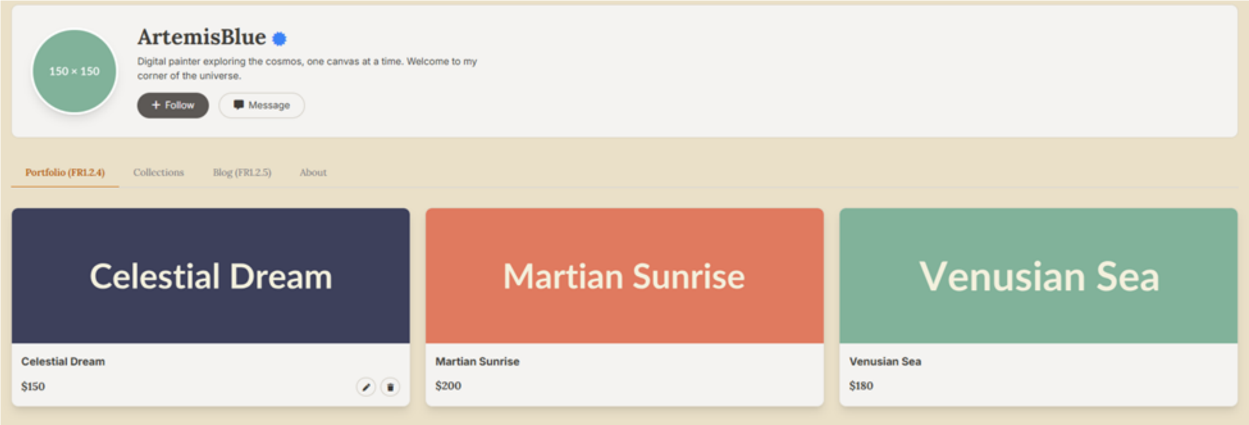


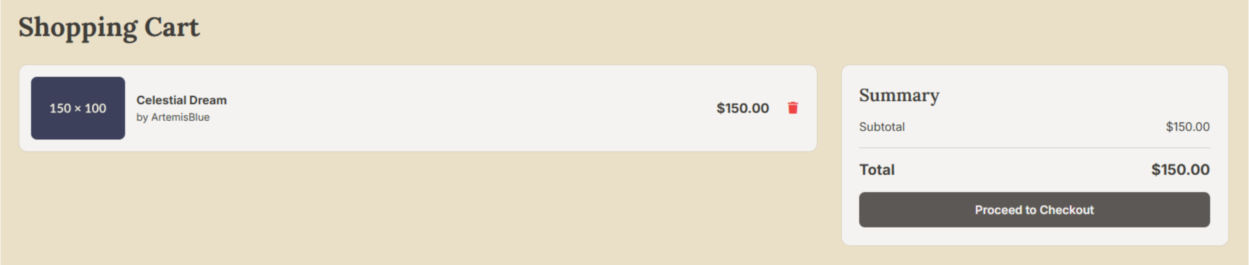
Fig.8: Upload Artwork



Digital painter exploring the cosmos, one canvas at a time. Welcome to my corner of the universe.

Celestial Dream	$150
Martian Sunrise	$200
Venusian Sea	$180

Fig.9: Artist Portfolio

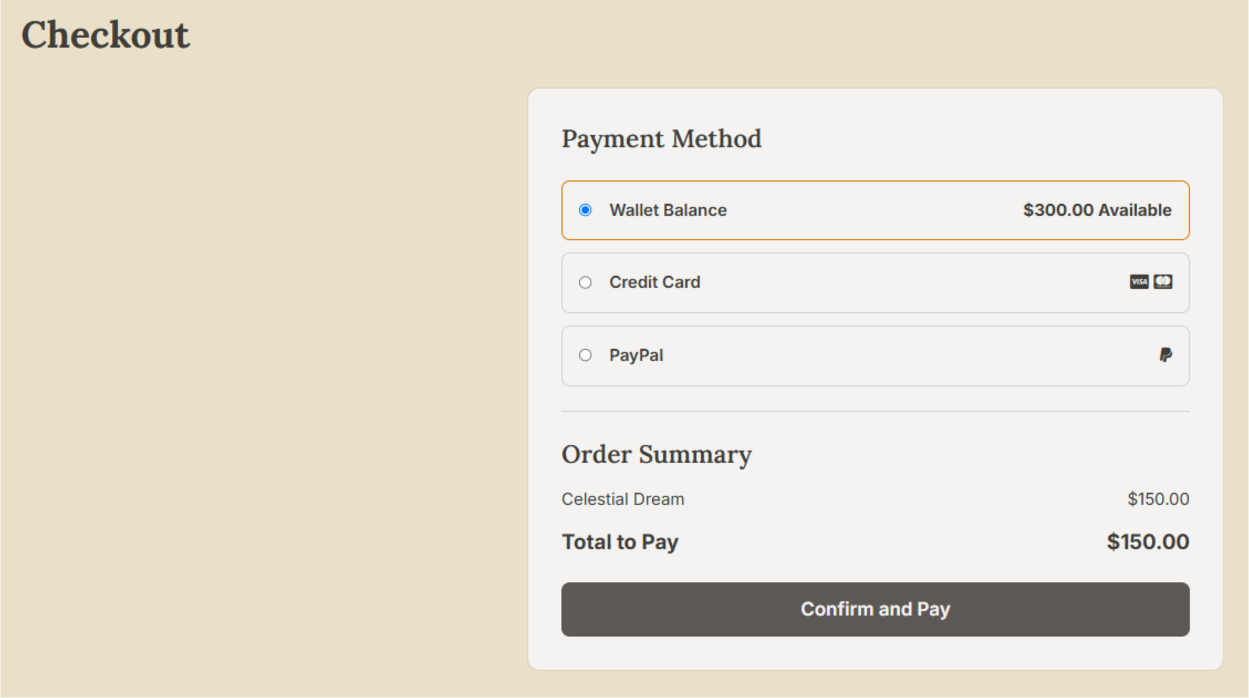


Celestial Dream
by ArtemisBlue
$150.00

Summary

Subtotal	$150.00
Total	$150.00

Fig.10: Shopping Cart



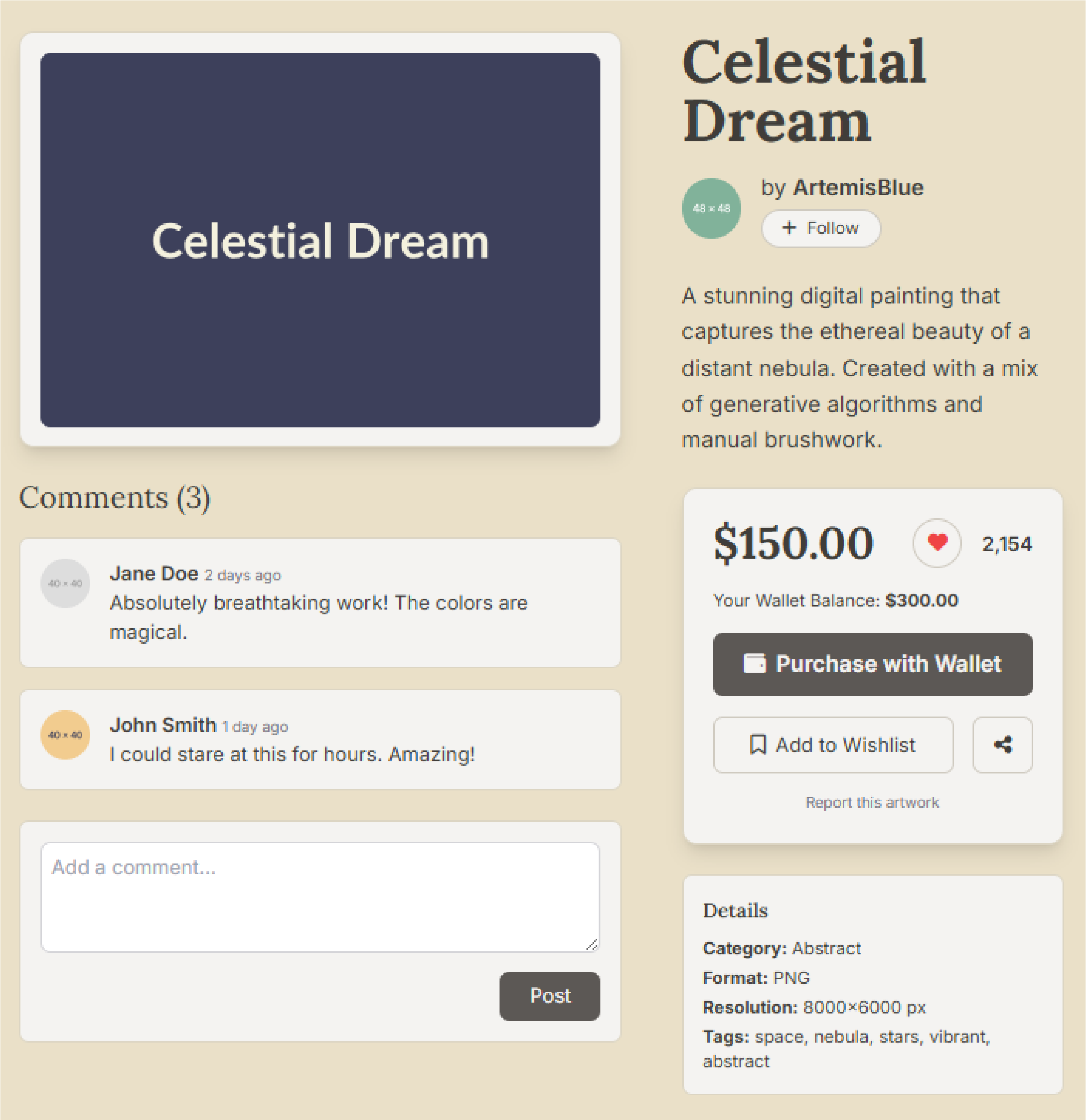
● Wallet Balance — $300.00 Available

○ Credit Card

○ PayPal

Celestial Dream	$150.00
Total to Pay	$150.00

Fig.11: Purchase Checkout (Wallet)



Celestial Dream

by ArtemisBlue

A stunning digital painting that captures the ethereal beauty of a distant nebula. Created with a mix of generative algorithms and manual brushwork.

Comments (3)

Jane Doe 2 days ago
Absolutely breathtaking work! The colors are magical.

John Smith 1 day ago
I could stare at this for hours. Amazing!

Add a comment...

$150.00 2,154

Your Wallet Balance: **$300.00**

Details

Category: Abstract
Format: PNG
Resolution: 8000×6000 px
Tags: space, nebula, stars, vibrant, abstract

Fig.12: Like, Comment, Follow

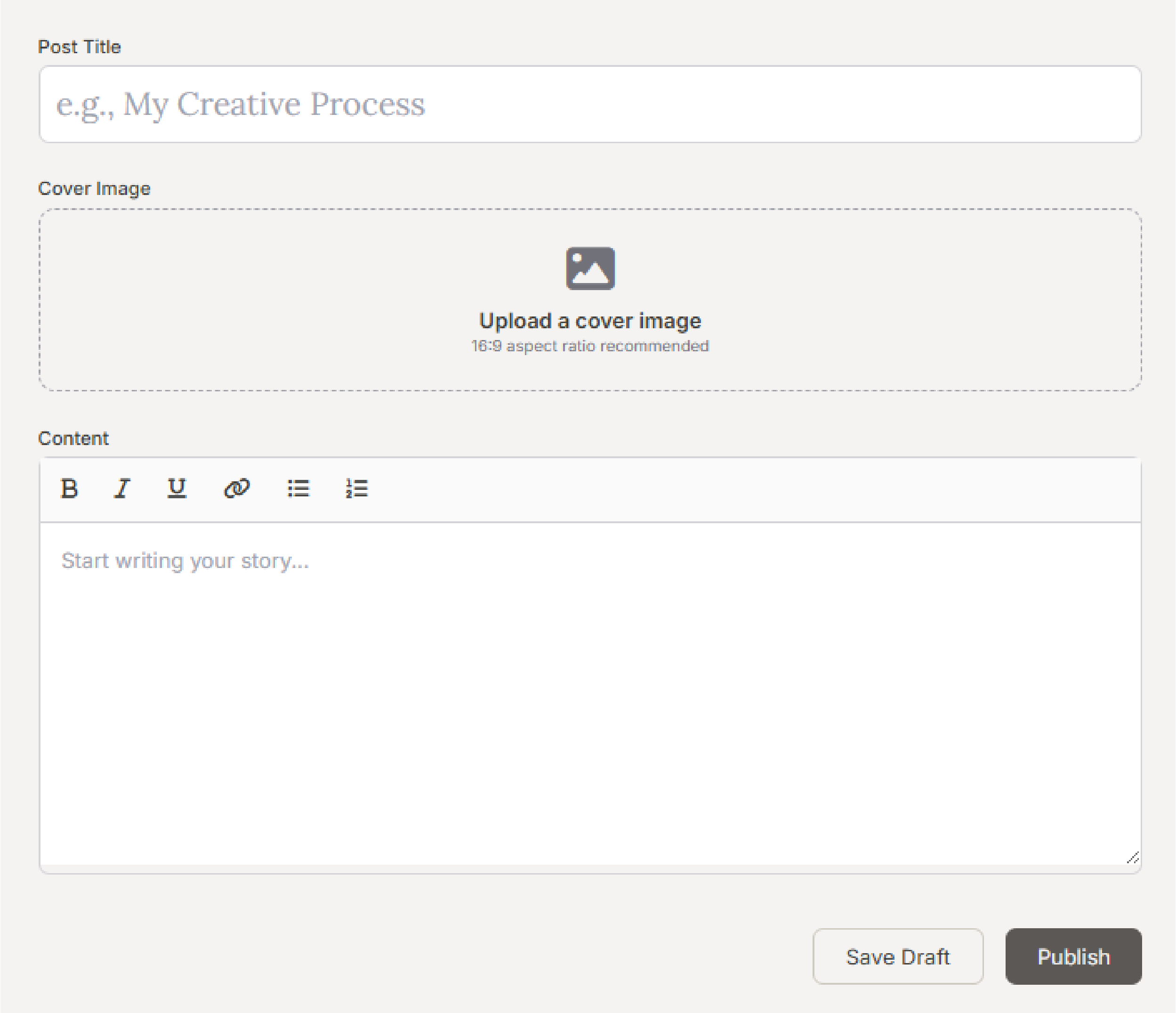
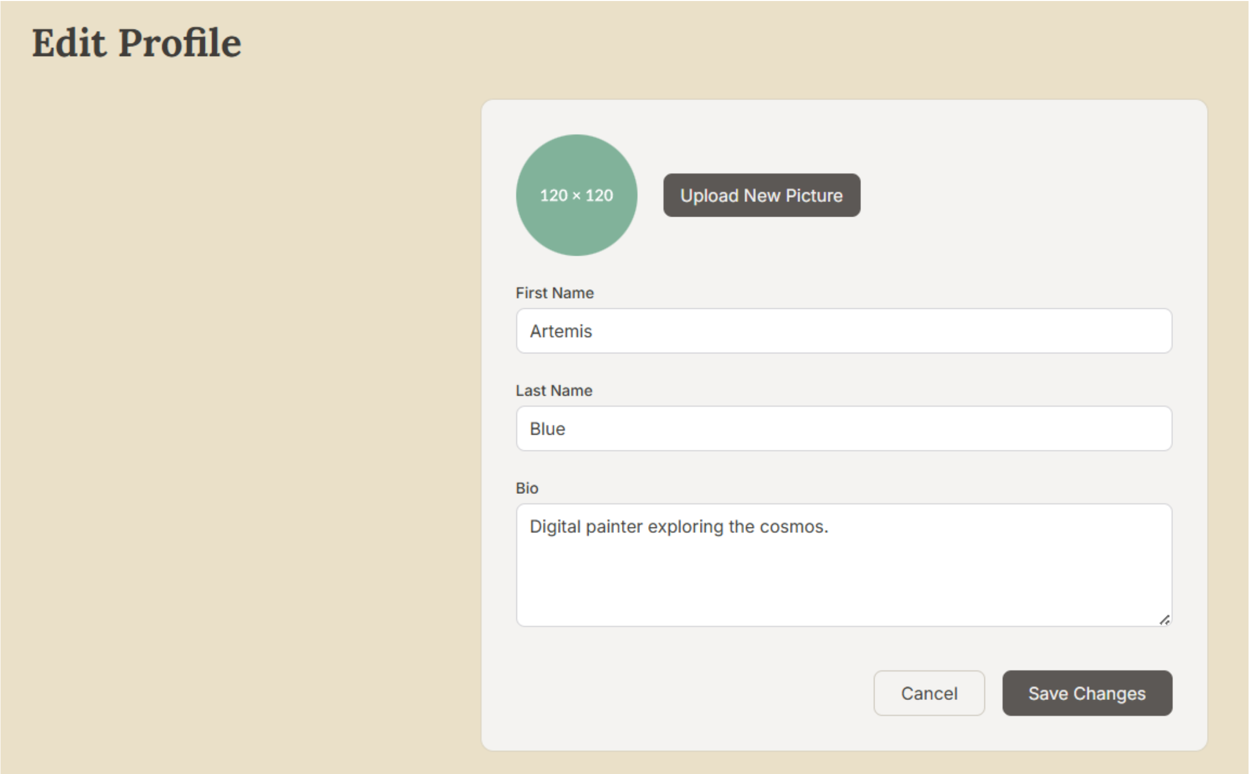

Post Title
e.g., My Creative Process
Cover Image
Upload a cover image
16:9 aspect ratio recommended
Content
B
I
U
Start writing your story...
Save Draft
Publish

Fig.13: Create Blog Post



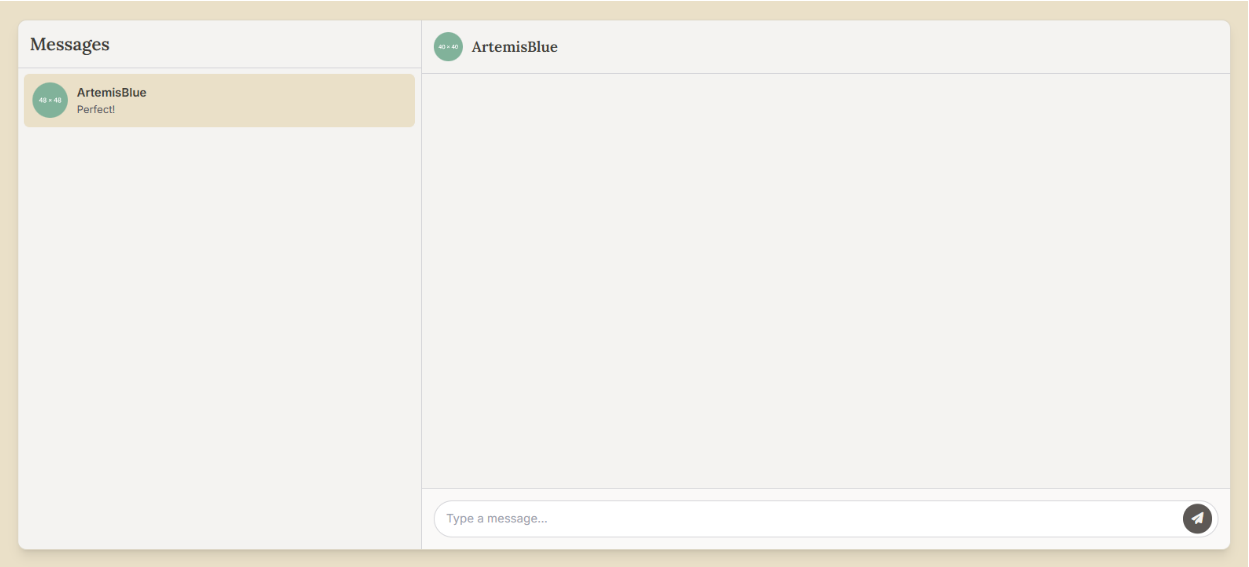
Edit Profile

First Name: Artemis

Last Name: Blue

Bio: Digital painter exploring the cosmos.

Fig.14: Edit Your Profile

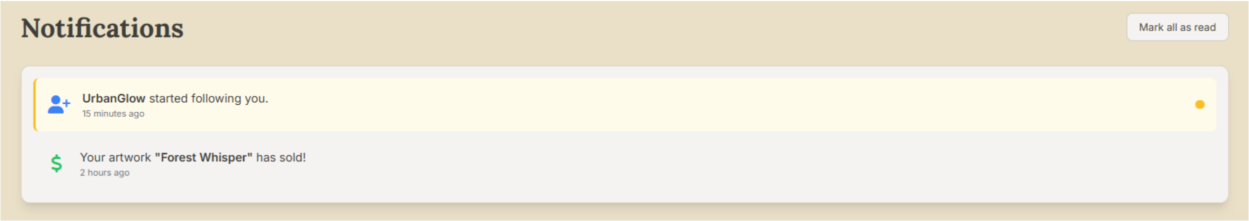


Messages

ArtemisBlue
Perfect!

ArtemisBlue

Fig.15: Messaging

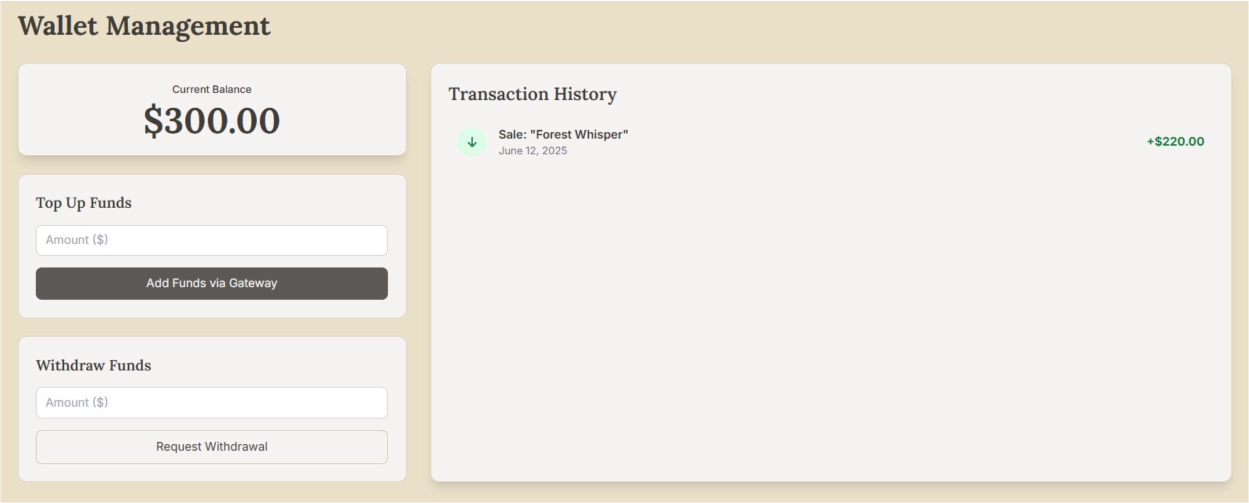


Notifications

UrbanGlow started following you.
15 minutes ago

Your artwork **"Forest Whisper"** has sold!
2 hours ago

Fig.16: Notifications



Wallet Management

Current Balance
$300.00

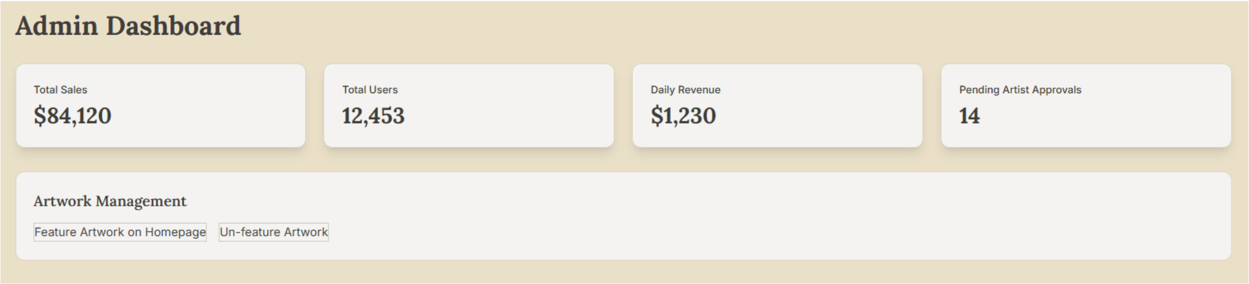
Top Up Funds
Amount ($)

Withdraw Funds
Amount ($)

Transaction History

Sale: "Forest Whisper" +$220.00
June 12, 2025

Fig.17: Wallet Management

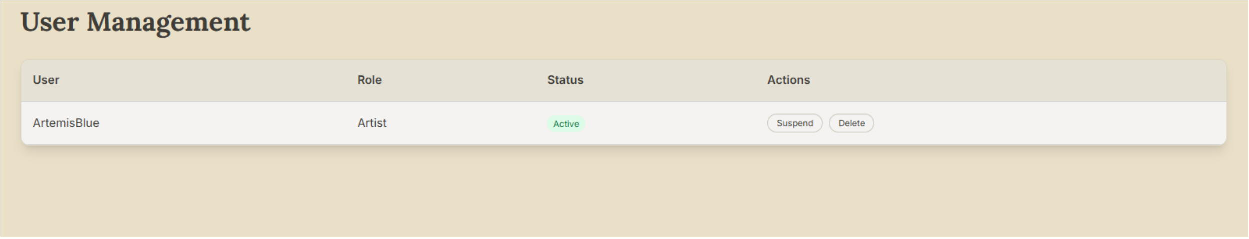


Admin Dashboard

Total Sales	Total Users	Daily Revenue	Pending Artist Approvals
$84,120	12,453	$1,230	14

Artwork Management

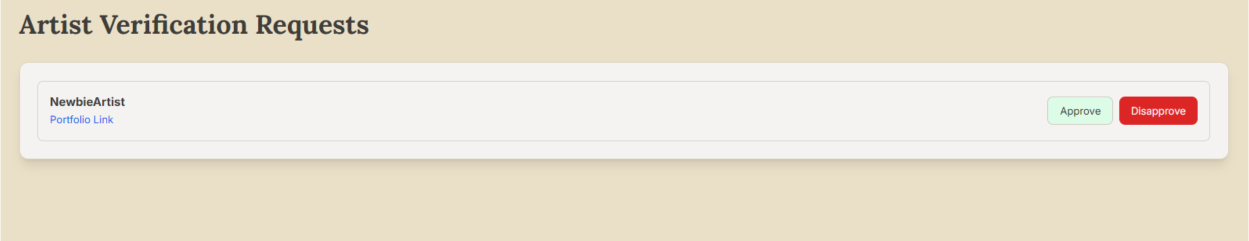
Fig.18: Admin Dashboard



User Management

User	Role	Status	Actions
ArtemisBlue	Artist	Active	Suspend Delete

Fig.19: User Management

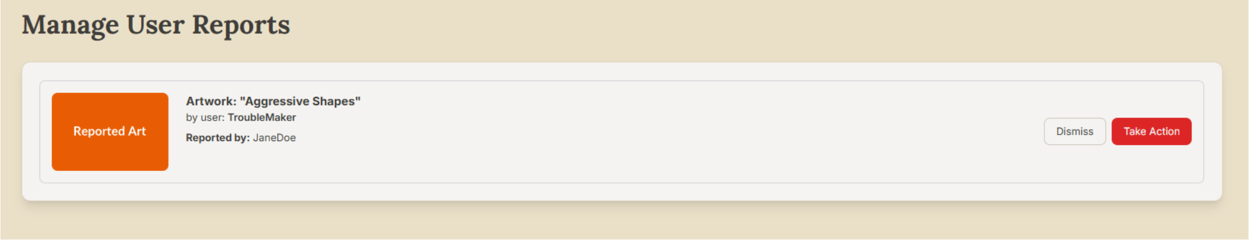


Artist Verification Requests

NewbieArtist

Portfolio Link

Fig.20: Artist Verification



Manage User Reports

Artwork: "Aggressive Shapes"

by user: TroubleMaker

Reported by: JaneDoe

Fig.21: User Report Management

4.4 Project Requirements

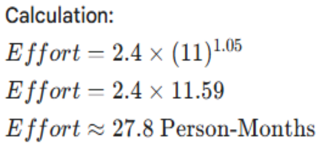
- **Time Estimation:**

We used COCOMO (constructive cost model) to calculate development time. Our Project is an Organic Project.

Software Project Type	Coefficient <Effort Factor>	P	T
Organic	2.4	1.05	0.38
Semi-Detached	3.0	1.12	0.35
Embedded	3.6	1.20	0.32

➢ Effort Estimation (COCOMO)

$$Effort = a \times (KLOC)^b$$



Calculation:

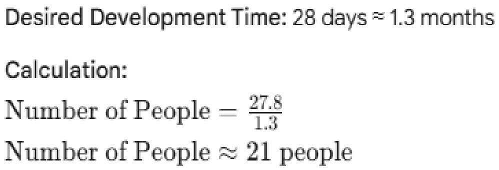
$Effort = 2.4 \times (11)^{1.05}$

$Effort = 2.4 \times 11.59$

$Effort \approx 27.8$ Person-Months

➢ Personnel Requirement

$$\text{Number of People} = \frac{\text{Total Effort}}{\text{Desired Development Time}}$$



Desired Development Time: 28 days ≈ 1.3 months

Calculation:

$\text{Number of People} = \frac{27.8}{1.3}$

$\text{Number of People} \approx 21$ people

Conclusion: To complete a project with an estimated effort of 27.8 person-months within the accelerated 28-day timeline, a team of approximately 21 people would be required.

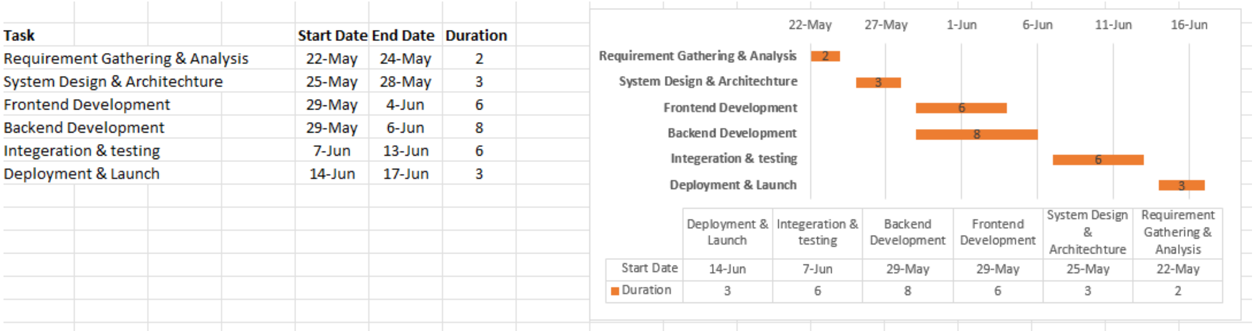
Time Estimated: 28 days (May 22, 2025 – July 17, 2025)

Scheduling Table:

Task	Predecessor	Duration (Days)	Resources Required
Requirement Gathering and analysis	-	2 days	• Business Analyst: 1 • Project Manager: 1
System Design and Architecture	-	3 days	• System Architect: 1 • Database Architect: 1 • UI/UX Designer: 1
Fronted Development	Requirement analysis and gathering and Designing the system and Architecture	6 days	• Fronted Developers (HTML/CSS/JS): 2
Backend Development	Gathering Requirement and analysis and System Design & Architecture	8 days	• Backend Developers (PHP,JavaScript): 2 • Database Administrator: 1
Integration and Testing	Frontend & Backend Development	6 days	• Quality Assurance Engineers: 3 • Frontend Developers: 2 • Backend Developers: 2
Deployment and Launch	Integration and Testing	3 days	• System Administrator: 1 • Marketing Team: 1
	Total=	28 days	21 People

- **Total Development Time:** 28 days

Gantt Chart-



Task	Start Date	End Date	Duration
Requirement Gathering & Analysis	22-May	24-May	2
System Design & Architechture	25-May	28-May	3
Frontend Development	29-May	4-Jun	6
Backend Development	29-May	6-Jun	8
Integeration & testing	7-Jun	13-Jun	6
Deployment & Launch	14-Jun	17-Jun	3

	Deployment & Launch	Integration & testing	Backend Development	Frontend Development	System Design & Architechture	Requirement Gathering & Analysis
Start Date	14-Jun	7-Jun	29-May	29-May	25-May	22-May
Duration	3	6	8	6	3	2

- **Budget Estimation**

  - **Personal Costs:**

    Project Manager: $50/day * 28 days = $1400

    Developers (3): $30/day * 4 developers * 14 days = $1680

    UI/UX Designer: $25/day * 3 days = $75 (assumed involvement for initial design)

    QA Tester: $30/day * 6 days = $180 (involved part-time initially, then full-time during testing)

  Total Personnel Cost = $3335

  - **Hardware and Software Costs**:

    Testing Devices (mobile phones, tablets for testing compatibility): $3,000

    Software Tools and Licenses (IDE licenses, testing tools, version control systems, etc.): $5,000

    Cloud Services (hosting, database, and deployment): $500/month * 0.93 months = $465

  Total Hardware and Software Cost = $8465

  - **Miscellaneous Costs**:

    Office Space and Utilities: $2,000/month * 0.93 months = $1860

    Training Costs (for team members on specific tools and technologies): $8,000

    Meetings (for client meetings, industry expert consultations): $200

    Contingency Budget (10% of total budget to handle unexpected expenses): $200

  Total Miscellaneous Cost = $10,260

  - **Total Project Budget:**

    Personnel Cost: $3335

    Hardware and Software Cost: $8465

    Miscellaneous Cost: $10260

  Total Estimated Budget: $ 22060

- **Resources**:

  Team Composition: The development team will consist of a project manager, 4 developers, 1 UI/UX designer, and 3 QA tester.

  Hardware and Software: Development will require standard computing equipment, development tools, and access to cloud services for deployment.

- **Environment**:

  Development Environment: The project will be developed using JavaScript and PHP for front-end, and a back-end system based on Node.js, using a cloud-hosted database.

  Testing Environment: Test servers will be set up to replicate the production environment for system and acceptance testing

- **Other Constraints**:

  The project must comply with data protection regulations (e.g., GDPR) to ensure user data is securely handled.

  Integration with third-party payment gateways must be completed and tested by Month 10 to allow sufficient time for performance validation.

5. FEATURES NOT TO BE TESTED

The following features will not be specifically addressed during the testing phase of the "Digital Canvas Hub" project-

- **Third-Party Payment Gateway Integration:**

  Detailed testing of third-party payment gateways such as PayPal, Bkash, Nagad and other third-party integrated payment methods will not be performed.

- **Infrastructure and Hosting:**

  We will not explicitly test the resilience and failover of cloud hosting services and server architecture. It's not this project's job to keep internet infrastructure maintained and tested.

- **Security Penetration and Vulnerability Testing:**

  Full security vulnerability scanning, penetration testing, and in-depth cybersecurity audits are out of the scope of this testing work. Simple security checks would also be indirectly tested using standard functional testing.

- **Performance under Real-World Internet Conditions:**

  No specific testing on system under various and bad network conditions (e.g., high latency, packet loss) will be performed.

6. TESTING APPROACH

6.1 Testing Levels

The testing process for the web-based platform “Digital Canvas Hub” will follow a multi-level strategy to ensure the delivery of a reliable, user-friendly, and functionally complete product. The testing method follows the standard software testing practices and is crafted and designed to the web application's modular design and user-focused functionality.

Unit Testing: Unit Testing particularly aims to validate individual components or functions created within the application system. This include verifying specific operations like input checking or validating, file management, wallet transaction logic, or blog content. In the early phases of development, developers will carry out foundational unit testing for a project like Digital Canvas Hub using White Box testing techniques. As this is the initial stage of the testing cycle, it ensures each software unit functions correctly before integration.

Integration Testing: Integration testing will verify different modules of the platforms such as user registration, artwork upload, messaging, wallet operations, and admin interactions by interoperating seamlessly. By the following completion of unit testing, components will be incrementally combined and tested to verify module to module communication. This helps detect interface errors and inconsistencies across connected services. The objective is to gradually assemble a stable web application and confirm that newer modules do not negatively impact on the performance of previously integrated ones. Integration testing will ensure completeness when the full system is assembled, test cases have been run, and critical/moderate bugs and defects are resolved.

System Testing: System testing will validate the performance of the fully integrated Digital Canvas Hub platform against both functional and non-functional requirements. Testing features like user profile updates, artwork purchases, search filters, and overall response of the system are also included. System testing reflects how end users would experience the platform and is performed by independent QA testers using black box testing techniques. It is a critical stage in the development process, especially as delivery deadlines come to an end. The layer ensures that defects must be fixed which don’t introduce regression and perform reliably across the system.

Acceptance Testing: Acceptance testing is the final level where the fully integrated application is evaluated to determine whether it meets stakeholder requirements and is ready for a lunch of deployment. This testing phase will require simulated use by target users (artists, buyers, and admins) to validate usability, accuracy, and completeness of key functions like artwork publishing, wallet-based purchasing, and content interaction. Acceptance testing validates the software is almost defect-free state to proceed to live functions. The final objective is to confirm the system meets agreed-upon criteria, requirements, identifying any remaining defects, and verify readiness for release.

6.2 Test Tools

For system interface designing, Figma is used to make a prototype of "Digital Canvas Hub" at first. And for backend and frontend development, these languages are used- Html, CSS, PHP, JavaScript, MySQL. For that, the software tools are- Visual studio code, XAMMP, GitHub. As for the testing for this system, Selenium will be used in terms of automated testing. Other than manual testing, Selenium will be of great use for enabling tests of web functionalities and quality attributes assurance while maintaining interface and feature performance together.

6.3 Meetings

First Meeting: A primary meeting is held before anything of the project "Digital Canvas Hub" to decide on goals, requirements, team roles, deliverables and dates, budget etc. This meeting includes the stakeholder/client, business analyst, project manager, system architect and database architect to ensure and match the requirements, expected result and deals.

Weekly Development Meeting: A development team meeting is held once every week. The meeting is conducted among the project manager, frontend developers, backend developers, UI/UX designer, database administrator and one quality assurance engineer to monitor the work progress and task updates over the week, identify any problem, discuss upcoming tasks.

Testing Review Meeting: After every testing phase like unit testing, integration testing and system testing- meetings are held among the quality assurance engineers, frontend and backend developers, system administrator. Also, during Acceptance testing, the project manager and client side must be present. This test review meeting is important for identifying and fixing bugs and defects, calling for retesting, performance and requirements fulfilment checks.

Project Handover Meeting: A final meeting is set up for handing over the project to the client. The client, project manager, marketing team, system architect should be present at this meeting explaining the overall project deliverable, receiving feedback and discussing maintenance.

7. TEST CASES/TEST ITEMS

Test Case-1

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Md. Asheke Rabbi</td></tr>
<tr><td colspan="3">Test Case ID: FR_REG_01</td><td colspan="2">Test Designed date: 29/05/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Registration</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify successful registration with valid data for a "User" account</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test case ensures a new user can create a "User" account successfully</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition : The user is on the registration page (Fig.2) and has a valid, unused email address</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Select the "User" account type.
2. Enter a unique username, valid email, and secure password.
3. Click "Register"</td><td>Account Type: User
Username: asheke1102
Email:asheketest@gmail.com
Password: ashekEE11@</td><td>"User" option is selected.
The system accepts all inputs without inline errors.
The user is redirected to an email verification page, and a confirmation email is sent to their address</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: A new user account is created in the database with the "User" role and an 'inactive' status, awaiting email activation..</td></tr>
</table>

Table 1: Test Case for **Registration as a User**

Test Case-2

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Md. Asheke Rabbi</td></tr>
<tr><td colspan="3">Test Case ID: FR_REG_02</td><td colspan="2">Test Designed date: 29/05/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Registration</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify successful registration with valid data for an "Artist" account.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test case ensures a new user can create an "Artist" account successfully.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition : The user is on the registration page (Fig.2) and has a valid, unused email address.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Select the "Artist" account type.
2. Enter a unique username, valid email, and secure password.
3. Click "Register"</td><td>Account Type: Artist
Username: asheke1102
Email:asheketest@gmail.com
Password: ashekEE11@</td><td>"Artist" option is selected.
The system accepts all inputs without inline errors.
The user is redirected to an email verification page, and a confirmation email is sent to their address</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: A new user account is created in the database with the "Artist" role and an 'inactive' status, awaiting email activation..</td></tr>
</table>

Table 2: Test Case for **Registration as an "Artist"**

Test Case-3

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Md. Asheke Rabbi</td></tr>
<tr><td colspan="3">Test Case ID: FR_REG_03</td><td colspan="2">Test Designed date: 30/05/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Registration</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify registration fails when using a username that is already taken.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test validates that all usernames must be unique.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition: A user with the username new_artist_101 already exists.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Fill out the registration form.
2. Click "Register"</td><td>Username: asheke1102
-</td><td>An inline error message appears below the username field stating, "This username is already taken. Please choose another."
The registration process is halted, and the user remains on the page with the error visible.</td><td>Not expected</td><td>Fail</td></tr>
<tr><td colspan="5">Post Condition: No new user account has been created in the database. The system state remains unchanged.</td></tr>
</table>

Table 3: Test Case for **Registration Attempt with Existing Username**

Test Case-4

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Md Asheke Rabbi</td></tr>
<tr><td colspan="3">Test Case ID: FR_REG_04</td><td colspan="2">Test Designed date: 30/5/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Registration</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify registration fails when using an invalid email format.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test checks the system's email format validation.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition : User is on the registration page.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Fill out the registration form.
2. Click "Register"</td><td>Email: invalid-email
-</td><td>An inline error message appears below the email field stating, "Please provide a valid email."
The registration process is halted.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: No new user account has been created in the database. The system state remains unchanged.</td></tr>
</table>

Table 4: Test Case for **Registration Attempt with Invalid Email Format**

Test Case-5

Project Name: Digital Canvas Hub			Test Designed by: Md Asheke Rabbi	
Test Case ID: FR_REG_05			Test Designed date: 31/05/25	
Test Priority (Low, Medium, High): High			Test Executed by: Tanvir Arafat Sahil	
Module Name: Registration			Test Execution date: 08/06/25	
Test Title: Verify registration fails when the password does not meet complexity requirements.				
Description: This test validates the password strength policy.				
Precondition : User is on the registration page.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Fill out the registration form. 2. Click "Register"	Password: pass -	An inline error message appears stating the password requirements (8+ characters, uppercase, lowercase, number, special character). The registration process is halted.	As expected	Pass
Post Condition: No new user account is created in the database. The system state remains unchanged.				

Table 5: Test Case for **Registration Attempt with Weak Password**

Test Case-6

Project Name: Digital Canvas Hub	Test Designed by: Md. Asheke Rabbi
Test Case ID: FR_ LOG_01	Test Designed date: 31/05/25
Test Priority (Low, Medium, High): High	Test Executed by: Tanvir Arafat Sahil
Module Name: System Login	Test Execution date: 08/06/25
Test Title: Verify users can log in using correct credentials.	
Description: This test validates the primary login functionality for a registered and activated user.	
Precondition: The user has a registered and activated account.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the login page (Fig.4)		The login form is displayed.	As expected	Pass
2. Enter valid email and password.	Email: artist@example.com, Password: ValidPassword123!			
3. Click "Sign in".		The user is redirected to the OTP verification page (Fig.3) to complete the login.		

Post Condition: The system has validated the primary credentials and is now awaiting OTP verification to create a user session.

Table 6: Test Case for **Login with Correct Credentials**

Test Case-7

Project Name: Digital Canvas Hub			Test Designed by: Yeameen Aziz Abdullah	
Test Case ID: FR_ LOG_02			Test Designed date: 31/05/25	
Test Priority (Low, Medium, High): High			Test Executed by: Tanvir Arafat Sahil	
Module Name: System Login			Test Execution date: 08/06/25	
Test Title: Verify login fails with an incorrect password.				
Description: This test ensures the system denies access for invalid password attempts.				
Precondition: The user has a registered and activated account.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the login page. 2. Enter valid email and incorrect password. 3. Click "Sign in".	Email: artist@example.com, Password: Incorrect Password	The login form is displayed. An error message "Invalid username or password" is displayed. The user remains on the login page.	As expected	Pass
Post Condition: A failed login attempt is logged against the user's account in the database. No user session is created.				

Table 7: Test Case for **Login with Incorrect Password**

Test Case-8

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Yeameen Aziz Abdullah</td></tr>
<tr><td colspan="3">Test Case ID: FR_ LOG_03</td><td colspan="2">Test Designed date: 01/06/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: System Login</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify that a CAPTCHA appears after 3 failed login attempts.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test validates the security feature to prevent brute-force attacks.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition: The user has a registered and activated account.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Enter incorrect login credentials three times.
2. Attempt a fourth login.</td><td>Email: artist@example.com, Password: wrongpassword
-</td><td>After the third failed attempt, the login page reloads to include a CAPTCHA challenge (Fig.5)
The user cannot proceed without successfully solving the CAPTCHA.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The system requires CAPTCHA validation for any subsequent login attempts for this user for a set period.</td></tr>
</table>

Table 8: Test Case for **Login Attempts Leading to CAPTCHA**

Test Case-9

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Yeameen Aziz Abdullah</td></tr>
<tr><td colspan="3">Test Case ID: FR_ LOG_04</td><td colspan="2">Test Designed date: 02/06/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: System Login</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify that the user's account is locked after 5 failed login attempts.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test validates the account lockout security policy.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition : The user has already failed 4 login attempts.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Make a fifth unsuccessful login attempt.
2. Immediately try to log in with correct credentials.</td><td>Email: artist@example.com, Password: anotherwrongpassword
-</td><td>The system displays a message: "Account will be locked for 30 mins after 5 failed attempts” or a similar lockout message for one hour
Login is denied, and the lockout message remains.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The user's account is flagged as 'locked' in the database with a timestamp for when the lock expires.</td></tr>
</table>

Table 9: Test Case for **Account Lockout After 5 Failed Attempts**

Test Case-10

Project Name: Digital Canvas Hub			Test Designed by: Yeameen Aziz Abdullah	
Test Case ID: FR_ LOG_05			Test Designed date: 02/06/25	
Test Priority (Low, Medium, High): High			Test Executed by: Tanvir Arafat Sahil	
Module Name: System Login			Test Execution date: 08/06/25	
Test Title: Verify users can reset their password using the "Forgot Password" feature.				
Description: This test ensures the password recovery mechanism is functional.				
Precondition: The user has a registered and activated account.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. On the login page, click "Forgot Password?". 2. Enter the registered email address and click "Send Reset Link".	Email: your.email@example.com	The user is taken to the "Reset Password" page (Fig.6) A confirmation message is shown, and a password reset link is sent to the email address.	As expected	Pass
Post Condition: A unique, single-use password reset token is generated and associated with the user's account in the database.				

Table 10: Test Case for **Password Reset via "Forgot Password" Link**

Test Case-11

Project Name: Digital Canvas Hub	Test Designed by: Yeameen Aziz Abdullah
Test Case ID: FR_ART_01	Test Designed date: 02/06/25
Test Priority (Low, Medium, High): High	Test Executed by: Tanvir Arafat Sahil
Module Name: Sell Artwork	Test Execution date: 08/06/25
Test Title: Verify an artist can upload a new artwork with all required details.	
Description: This test validates the core artwork uploading feature for artists.	
Precondition : User is logged in with a verified "Artist" account.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the "Upload New Artwork" page (Fig.8).		The upload form is displayed.	As expected	Pass
2. Upload a file of a valid format (JPG, PNG) and size (under 50MB).	File: celestial_dream.png	A preview of the image appears.		
3. Fill in the title, description, price, and tags.	Title: Celestial Dream, Price: $150, etc.			
4. Click "Publish Artwork".		The artwork is published and visible in the artist's portfolio		
Post Condition: A new artwork record is created in the database, linked to the artist's profile, and the artwork file is stored on the server.				

Table 11: Test Case for **Artist Uploads New Artwork Successfully**

Test Case-12

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Yeameen Aziz Abdullah</td></tr>
<tr><td colspan="3">Test Case ID: FR_ART_02</td><td colspan="2">Test Designed date: 02/06/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): Medium</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Sell Artwork</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify artwork upload fails if the file size exceeds the 50MB limit.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test validates the file size constraint on uploads.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition : User is logged in with a verified "Artist" account.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Navigate to the "Upload New Artwork" page.
2. Attempt to upload a file larger than 50MB.</td><td>File: large_artwork.jpg (Size: 60MB)</td><td>The upload form is displayed.
The upload fails, and an error message is displayed: "File size exceeds the 50MB limit."</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: No artwork record is created, and no file is saved to the server. The system state remains unchanged.</td></tr>
</table>

Table 12: Test Case for **Artwork Upload Fails Due to Large File Size**

Test Case-13

Project Name: Digital Canvas Hub	Test Designed by: Nabiha Tahsin
Test Case ID: FR_ART_03	Test Designed date: 02/06/25
Test Priority (Low, Medium, High): Medium	Test Executed by: Tanvir Arafat Sahil
Module Name: User Profile Management	Test Execution date: 08/06/25
Test Title: Verify an artist can remove or un-list their own artwork.	
Description: This test validates that artists have control over their submitted works.	
Precondition : An artist is logged in and is viewing their own portfolio.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the artist's portfolio page.	-	The gallery of uploaded artworks is displayed.	As expected	Pass
2. Select an artwork to delete.	-	Management options for the artwork are visible.		
3. Click the "Remove" or "Delete" option and confirm the action.	-	The artwork is no longer visible in the portfolio or on the public site.		
Post Condition: The artwork's status is changed to 'deleted' or the record is removed from the database.				

Table 13: Test Case for **Artist Deletes Their Own Artwork**

Test Case-14

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Nabiha Tahsin</td></tr>
<tr><td colspan="3">Test Case ID: FR_ PUR_01</td><td colspan="2">Test Designed date: 03/06/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Purchase Artwork</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify a user can successfully purchase an artwork using their wallet.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test validates the end-to-end purchasing workflow.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition : The user is logged in and has sufficient funds in their wallet.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Navigate to an artwork page (Fig.12) and click "Purchase with Wallet".
2. Verify the order summary and click "Confirm and Pay".</td><td>Artwork Price: $150, Wallet Balance: $300
-</td><td>The user is taken to the checkout page (Fig.11).
The purchase is successful. The user's wallet balance is deducted by $150. A success notification is shown.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The transaction is recorded in the transaction histories of both buyer and seller. The buyer's wallet is debited, and the seller's wallet is credited.</td></tr>
</table>

Table 14: Test Case for **User Purchases Artwork with Wallet Balance**

Test Case-15

Project Name: Digital Canvas Hub	Test Designed by: Nabiha Tahsin
Test Case ID: FR_ PUR_02	Test Designed date: 03/06/25
Test Priority (Low, Medium, High): Medium	Test Executed by: Tanvir Arafat Sahil
Module Name: Purchase Artwork	Test Execution date: 08/06/25
Test Title: Verify a user can add more than one artwork to the shopping cart.	
Description: This test validates the multi-item shopping cart functionality.	
Precondition : User is logged in.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Add a first artwork to the cart.	Artwork 1	The cart icon updates to show 1 item.	As expected	Pass
2. Browse to a second artwork and add it to the cart.	Artwork 2	The cart icon updates to show 2 items.		
3. View the shopping cart (Fig.10).	-	Both artworks are listed correctly with an updated subtotal and total.		
Post Condition: The user's session data for the shopping cart is updated to include both items.				

Table 15: Test Case for **User Adds Multiple Items to Shopping Cart**

Test Case-16

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Nabiha Tahsin</td></tr>
<tr><td colspan="3">Test Case ID: FR_ INT_01</td><td colspan="2">Test Designed date: 03/06/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): Medium</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Like, Comment & Follow</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify a logged-in user can like and comment on an artwork.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test validates the social interaction features on artwork pages.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition : User is logged in.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Navigate to an artwork page (Fig.12).
2. Click the heart/like icon.
3. Type a comment and click "Post".</td><td>-
-
Comment: Amazing work!</td><td>The like button and comment section are visible.
The like count increments by one.
The new comment appears publicly in the comments section.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: New records for the 'like' and 'comment' are created in the database and associated with the user and the artwork.</td></tr>
</table>

Table 16: Test Case for **User Likes and Comments on an Artwork**

Test Case-17

Project Name: Digital Canvas Hub			Test Designed by: Nabiha Tahsin	
Test Case ID: FR_ SRC_01			Test Designed date: 03/06/25	
Test Priority (Low, Medium, High): High			Test Executed by: Tanvir Arafat Sahil	
Module Name: Search and Discovery			Test Execution date: 08/06/25	
Test Title: Verify users can find other users and artists by searching for their username.				
Description: This test validates the user search functionality.				
Precondition : User is logged in. An artist named "Artemis Blue" exists.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Use the main search bar on the homepage. 2. Click the profile link.	Search Term: Artemis Blue -	The search results page displays a link to the "Artemis Blue" artist profile. The user is successfully navigated to the correct artist's portfolio page.	As expected	Pass
Post Condition: New records for the 'like' and 'comment' are created in the database and associated with the user and the artwork.				

Table 17: Test Case for **Search for an Artist by Username**

Test Case-18

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Nabiha Tahsin</td></tr>
<tr><td colspan="3">Test Case ID: FR_ SRC_02</td><td colspan="2">Test Designed date: 04/06/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): Medium</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Search and Discovery</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify the "Trending Art" section on the homepage displays relevant content.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test ensures that discovery features on the homepage are functional.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition: User is visiting the homepage.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Navigate to the homepage.
2. Click on an artwork within the "Trending Art" section.</td><td>-
-</td><td>The homepage displays a section titled "Trending Art"
The user is correctly redirected to the artwork's detail page.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The user is on the artwork detail page. The click action is logged for analytics.</td></tr>
</table>

Table 18: Test Case for **Using "Trending Art" Discovery Feature**

Test Case 19

Project Name: Digital Canvas Hub			Test Designed by: Yeameen Aziz Abdullah	
Test Case ID: FR_ ADM_01			Test Designed date: 04/06/25	
Test Priority (Low, Medium, High): High			Test Executed by: Tanvir Arafat Sahil	
Module Name: Admin Functions			Test Execution date: 08/06/25	
Test Title: Verify the admin can access site-wide analytics on the dashboard.				
Description: This test ensures the admin dashboard correctly displays key site metrics.				
Precondition: User is logged in with admin-level credentials.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Admin Dashboard (Fig.18).	-	The dashboard correctly displays statistics for "Total Sales," "Total Users," and "Daily Revenue".	As expected	Pass
Post Condition: The admin has successfully viewed the site-wide data. No data is changed.				

Table 19: Test Case for **Admin Views Site-Wide Statistics**

Test Case 20

<table>
<tr><td colspan="3">Project Name: Digital Canvas Hub</td><td colspan="2">Test Designed by: Yeameen Aziz Abdullah</td></tr>
<tr><td colspan="3">Test Case ID: FR_ ADM_02</td><td colspan="2">Test Designed date: 04/06/25</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="3">Module Name: Admin Functions</td><td colspan="2">Test Execution date: 08/06/25</td></tr>
<tr><td colspan="3">Test Title: Verify an admin can feature an artwork to appear on the homepage.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: This test validates the admin's content curation capabilities.</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition: User is logged in with admin-level credentials.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Navigate to the "Artwork Management" section of the admin dashboard.</td><td>-</td><td>The admin can search for or view artworks.</td><td>As expected</td><td>Pass</td></tr>
<tr><td>2. Select an artwork and click "Feature Artwork on Homepage".</td><td>-</td><td>A success message is displayed.</td><td></td><td></td></tr>
<tr><td>3. Navigate to the main homepage.</td><td>-</td><td>The selected artwork now appears in the "Featured Artists" or a similar featured section.</td><td></td><td></td></tr>
<tr><td colspan="5">Post Condition: The artwork's 'is_featured' status is updated to true in the database.</td></tr>
</table>

Table 20: Test Case for **Admin Features an Artwork on the Homepage**

Test Case-21

Project Name: Digital Canvas Hub	Test Designed by: Yeameen Aziz Abdullah
Test Case ID: FR_ ADM_03	Test Designed date: 04/06/25
Test Priority (Low, Medium, High): High	Test Executed by: Tanvir Arafat Sahil
Module Name: Admin Functions	Test Execution date: 08/06/25
Test Title: Verify an admin can suspend a user account from the user management panel.	
Description: This test validates the admin's ability to suspend user accounts.	
Precondition: User is logged in with admin credentials.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the "User Management" page (Fig.19).	-	A list of users is displayed.	As expected	Pass
2. Find the user "Artemis Blue" and click "Suspend".	-	A confirmation dialog appears.		
3. Confirm the suspension.	-	The user's status changes to "Suspended". The suspended user can no longer log in.		
Post Condition: The user's account status is updated to 'suspended' in the database.				

Table 21: Test Case for **Admin Suspends a User Account**

Test Case-22

Project Name: Digital Canvas Hub			Test Designed by: Yeameen Aziz Abdullah	
Test Case ID: FR_ NFR_01			Test Designed date: 04/06/25	
Test Priority (Low, Medium, High): High			Test Executed by: Tanvir Arafat Sahil	
Module Name: Reliability			Test Execution date: 08/06/25	
Test Title: Verify that the database is automatically backed up nightly.				
Description: This test verifies the critical data backup and recovery process.				
Precondition: Access to server logs and the backup storage location is required.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Check the server logs for the previous night at the scheduled backup time. 2. Navigate to the backup storage location (e.g., cloud storage bucket).	- -	A log entry confirms that the automated backup script was executed successfully. A new database backup file exists with a timestamp corresponding to the last execution.	As expected	Pass
Post Condition: A complete and valid database backup file is securely stored.				

Table 22: Test Case for **Reliability: Database Backup**

Test Case-23

<table>
<tr><td colspan="4">Project Name: Digital Canvas Hub</td><td>Test Designed by: Yeameen Aziz Abdullah</td></tr>
<tr><td colspan="4">Test Case ID: FR_ NFR_02</td><td>Test Designed date: 05/06/25</td></tr>
<tr><td colspan="4">Test Priority (Low, Medium, High): High</td><td>Test Executed by: Tanvir Arafat Sahil</td></tr>
<tr><td colspan="4">Module Name: Usability</td><td>Test Execution date: 08/06/25</td></tr>
<tr><td colspan="4">Test Title: Verify a new user can register and purchase an artwork in an average of five minutes.</td><td></td></tr>
<tr><td colspan="4">Description: This test measures the efficiency of the core user journey for new users.</td><td></td></tr>
<tr><td colspan="5">Precondition: A test participant who has never used the website is available.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Start a timer and ask the user to sign up, find an artwork, and purchase it.</td><td>-</td><td>-</td><td>As expected</td><td>Pass</td></tr>
<tr><td>2. Observe the user's interaction with the UI without providing help.</td><td>-</td><td>-</td><td></td><td></td></tr>
<tr><td>3. Stop the timer upon successful purchase confirmation.</td><td>-</td><td>The total time taken should be around five minutes, and the workflow should feel intuitive to the user.</td><td></td><td></td></tr>
<tr><td colspan="5">Post Condition: The test participant has successfully created an account and completed a purchase. Timings and qualitative feedback are recorded.</td></tr>
</table>

Table 23: Test Case for **Usability: New User Purchase Workflow Time**

Test Case-24

Project Name: Digital Canvas Hub			Test Designed by: Yeameen Aziz Abdullah	
Test Case ID: FR_ NFR_03			Test Designed date: 05/06/25	
Test Priority (Low, Medium, High): High			Test Executed by: Tanvir Arafat Sahil	
Module Name: Performance			Test Execution date: 08/06/25	
Test Title: Verify that web pages load completely in under 3 seconds with 500 concurrent users.				
Description: This test measures site performance under a simulated average load.				
Precondition: A load testing tool (e.g., JMeter) is configured to simulate 500 users.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Run the load test simulating 500 concurrent users Browse key pages. 2. Monitor the 95th percentile page load time metric from the tool's results.	Homepage, Search, Artwork pages -	- The page load time must remain below the 3-second threshold.	As expected	Pass
Post Condition: The performance test completes, and detailed logs are generated for analysis.				

Table 24: Test Case for **Performance: Web Page Load Time**

Test Case-25

Project Name: Digital Canvas Hub			Test Designed by: Yeameen Aziz Abdullah	
Test Case ID: FR_ NFR_04			Test Designed date: 06/06/25	
Test Priority (Low, Medium, High): High			Test Executed by: Tanvir Arafat Sahil	
Module Name: Scalability			Test Execution date: 08/06/25	
Test Title: Verify the system can handle a 50% increase in user traffic without performance degradation.				
Description: This test checks the system's ability to scale horizontally to manage traffic spikes.				
Precondition: A load testing environment is configured. The baseline performance at 500 users is known.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Run a load test with 750 concurrent users (a 50% increase from the average load of 500).	750 concurrent users	-	As expected	Pass
2. Monitor page load times and server resource utilization (CPU, RAM).	-	Page load times should not significantly degrade from the baseline, and server resources should not be maxed out, indicating the system has scaled effectively.		
Post Condition: The scalability test completes. Logs are generated, proving the system's ability to handle the increased load.				

Table 25: Test Case for **Scalability: Handling Increased User Traffic**

8. ITEM PASS/FAIL CRITERIA

Test Case-1: Validate the user should register with valid details when he is a "User".
Expected Result: The inputs are accepted with no inline errors and the user is taken to an email verification page.
Why It Should Pass: This will pass because it is valid data that the system should accept. We want the registration process to be smooth for actual users, who should be able to identify their own email and continue.

Test Case-2: Validate a new user registering successfully with correct details for 'Artist' type account.
Expected Result: Process all inputs with no inline errors, and redirect the user to an email verification page.
Why It Should Pass: This test will pass as artists should be able to create accounts just like any other user and with the same functionalities. We want artists to easily register and start uploading their work.

Test Case-3: Ensure that registration fails when a username is already in use.
Expected Result: An inline error message would display that reads, “This username is already in use. Please choose another."
Why It Should Fail: We want this to fail, because we expect the system to not allow a duplicate username. This is done to achieve uniqueness and prevent ambiguity among users. The error note explains what the issue was.

Test Case-4: Registration shouldn’t be possible with an invalid email (e.g., test@test).
Expected Result: The following in line error would be shown: "Please enter a valid email."
Why It Should Fail: The email format entered shouldn't be acceptable by the system. We want this to not work, as we want users to be forced to submit a valid email address for clear communication and account activation.

Test Case-5: Validate Failed Registration with Invalid Password. Validate that the registration process should fail if the user provides a password which is not of sufficient complexity.
Expected Result: An error message appears inline, communicating the password requirements

Why It Should Fail: We’d like it to fail because the system should be enforcing password readers! strength for security. Weak passwords become security holes in the system, so it is crucial to not accept passwords that fail the tests.

Test Case-6: Check whether users are able to log in with valid credentials.
Expected Result: User logs in using valid login details and is redirected to OTP verification page.
Why It Should Pass: This is because we would like the credentials to be valid for a successful login and be redirected to the OTP verification page to provide extra admin security. It makes sure the system properly fits logins.

Test Case-7: Confirm that you get login failed with incorrect password.
Expected Result: The error message "Invalid username or password" is shown.
Why It Should Fail: It's a security risk to accept a inavalid password. The security testifies that nobody but legitimate users can access the system, which means that unauthorized access is denied.

Test Case-8: To check whether CAPTCHA appear on 3 incorrect login attempts.
Expected Result: on the third failed attempt, the login page will become a CAPTCHA challenge.
Why It Should Pass: Because this test should pass to avoid brute forcing the system. It's a security layer to prevent non-humans from at least even making the login request (human is 4+3 symbols), pest also bot is unable to autologin with a 7 symbols long keyword.

Test Case-9: To ensure the user account will be locked if the number of failed attempts reaches 5.
Expected Result: The system shows a message: "Your account is locked during 30 min after 5 failed login attempts.
Why It Should Pass: This check should pass as account lockout is a security measure to protect against brute-force attacks. We're focused on making sure we secure users accounts so they can't be accessed by the wrong people.

Test Case-10: To check whether users are able to reset their password from “Forgot Password” link. **Expected Result:** A confirmation message is shown, and a password reset link is sent to the user’s email address. Using that link, user will be able to reset the password.
Why It Should Pass: This test should pass, as password recovery is essential. We want to make sure that if the user forgets their password, they can regain access to their accounts in a secure and user-friendly way.

Test Case-11: Check if an artist is able to upload a new artwork with all mandatory field details.
Expected Result: The artwork has been uploaded and is visible in the artist’s portfolio.
Why It Should Pass: This test should pass because artists should able to show their work spontaneously. Uploading successfully and being visible is essential for artist usage of the platform.

Test Case-12: Check acceptance of artwork if it is larger than 50MB.
Expected Result: Upload should fail and this error message should be shown: "The file size has exceeded the limit of 50MB."
Why It Should Fail: This test should fail because the system is designed with a maximum file size constraint of 50MB for artwork uploads to ensure optimal server performance, storage management, and user experience. Allowing uploads larger than this limit could cause server overload or timeout issues, lead to long upload times.

Test Case-13: Verify an artist should be able to delete or un-list their own artwork.
Expected Result: The artwork is deleted and no longer shows in the portfolio or public site.

Why It Should Pass: This test should pass because artists should have control over their own content. If an artist wants to remove or un-list an artwork, the system should support and enable that action.

Test Case-14: Check that a user will be able to buy artwork from the wallet successfully.
Expected Result: The purchase is successful, and the wallet's balance is updated, a success-alert will be shown.
Why It Should Pass: This test should pass because buying art via a wallet is a fundamental use case. We want the system to properly record transactions, update balances, and give the users confirmation of their successful purchases.

Test Case-15: To check that the user can add more than one artwork to the shopping cart.
Expected Result: The cart refreshes with all the other arts added and the right subtotal amount showing.
Why It Should Pass: This one should pass because the list of items in the user's cart is fundamental to the user experience and it must be in multi-item cart feature. We do not want to force people to add one artwork to the cart then check out.

Test Case-16: Check if a user is logged in then they can like and comment on artwork.
Expected Result The like and comment appear, and the like count increases by one.
Why It Should Pass: This one should pass because social interaction (likes, comments, etc.) is an integral component of the user experience. We want people to consume content, to engage, to join the community.

Test Case-17: Verify users can find other users and artists by searching for their username
Expected Result: The name of the artist comes up when searched.
Why It Should Pass: This test should pass to check if the search function is working perfectly

Test Case-18: Verify the "Trending Art" section on the homepage display relevant content.
Expected Result: If "Trending Art" of home screen shows relevant content or not.
Why It Should Pass: This test should pass because the "Trending Art" section is a discovery feature. We desire for people to experience popular art and end up in the right places.

Test Case-19: Verify the admin access site-wide analytics on the dashboard.
Expected Result: Admin will be able to see his statistics, like "Total Sales", "Total Users".
Why It Should Pass: This test should pass because the admin shall see performance and user data. We want a dashboard to convey the right analytics for decision support.

Test Case-20: Verify an admin can feature an artwork to appear on the homepage;
Expected Result: The "Featured Artists" section in the front page displays the piece of artwork.
Why It Should Pass: This test should pass since admins ought to be able to enhance the content of the home homepage. Displaying art is a way to highlight good, nice or fashionable art.

Test Case-21: Verify an admin can suspend a user and have the account status changed to "Suspended".

Expected Result: The user's status changes to "Suspended."
Why It Should Pass: This test should pass, because administrators should have control over user's accounts, including suspending an account (for example, when an account is suspended for moderation, or security reasons).

Test Case-22: Verify the database is backed up automatically.
Expected Result: A backup is created and logged each night.
Why It Should Pass: This test should be passed to make sure that the database is backed up frequently, lowering the risk of loss of data. Also, regular backups are important for system recovery.

Test Case-23: Check that a new user can sign up and buy a piece of art in 5 minutes.
Expected result: The person is registered and will be able to purchase an artwork within an average time of 5 minutes.
Why It Should Pass: This test should pass because we want that platform to be efficient and easy to use. Fast registration and purchase contribute also to improved user experience.

Test Case-24: Ensure web pages fully load within 3 seconds under 500 simultaneous users.
Expected Result: With 500 users, the web pages should be loaded within less than 3 seconds.
Why It Should Pass: This test should pass because performance is important. We want the system to handle traffic well, and quick enough page load times under heavy load of users that are active at the same time.

Test Case-25: Checking system's Scalability by handling the 50% increase in user traffic without lowering performance.
Expected Result: 750 users experienced no serious performance issue with the system.
Why It Should Pass: This test should pass as we would like the system to be able to scale. The fact that the platform can be scaled to accommodate high traffic without sacrificing performance also means the platform is ready for growth.

9. TEST DELIVERABLES

Test Plan Document: A brief description of the testing approach, objectives, scope, and schedule. It specifies what will be tested and how.

Test Cases: A test plan including a list of test cases that includes specific steps, expected results, and pass/fail expectation. Types of tests the following document is a "how-to" guide for testing.

Test Execution Logs: Actual results of the test run, with date, tester, and results (pass/fail) recorded. This is to indicate what tests were run.

Defect Reports: A report for any found issues or bugs, their severity, and how to reproduce them.

Test Summary Report: A summary of the testing, including passed and failed tests, and any key findings.

SRS (Software Requirements Specification): The document outlining the system's requirements. This helps testers understand the expected functionality of the system.

Code/Automated Test Scripts: Automated test scripts or code used for testing the system. When it comes to regression or performance testing it is very important.

Test Environment Setup: A description of the hardware, software, and tools used for testing, ensuring that the tests were performed in the correct environment.

10. STAFFING AND TRAINING NEEDS

The testing team for our project "Digital Canvas Hub" will be consist of professionals for every role to ensure a complete and comprehensive testing process. Expertise from various fields will be recruited and will assign in the project's task according to their roles and their relevant fields.

Staffing Needs:

1. **Quality Assurance (QA) Engineers: 3**
   - Role: Perform every test stage: Unit, Integration, System, Acceptance testing. QA will verify the platform met the needs of the functional and non-functional requirements.
2. **Front-End Developers: 2**
   - Roles: Front end developers will help to test the user interface (UIX) and connection of backend modules.
3. **Back-End Developers: 2**
   - Roles: Back-end developers will focus server-side logic of backend along with the database or API connection during testing.
4. **Ui/UX Designer: 2**
   - Roles: Performs the usability testing, modify the UI (user interface) as per requirements and verify its optimization.
5. **Database Administrator: 1**
   - Roles: Monitors the database scale, security, integrity, optimization and performance during testing.
6. **System Administrator: 1**
   - Roles: Monitors and manage the deployment and environment of the system for testing stages.
7. **Marketing Team: 1**
   - Roles: Manages the project's requirements or features, handle communication between clients, shares the testing results with stakeholders.

Training Needs:

- **Testing Tools Trainings:** Complete training on how to use Selenium for automated functional testing and how to use JMeter for performance and load testing
- **Technical Training:** Developers and testers will go through specific training in multiple languages according to our project PHP, JavaScript, HTML, CSS and MySQL along with the use of XAMMP for local environment.

- **Database Training:** Training in detail in database management with its performance, monitors, optimization, security and defects tracking.

- **UI Training:** Training in UI or user interface and usability testing techniques according to user requirements.

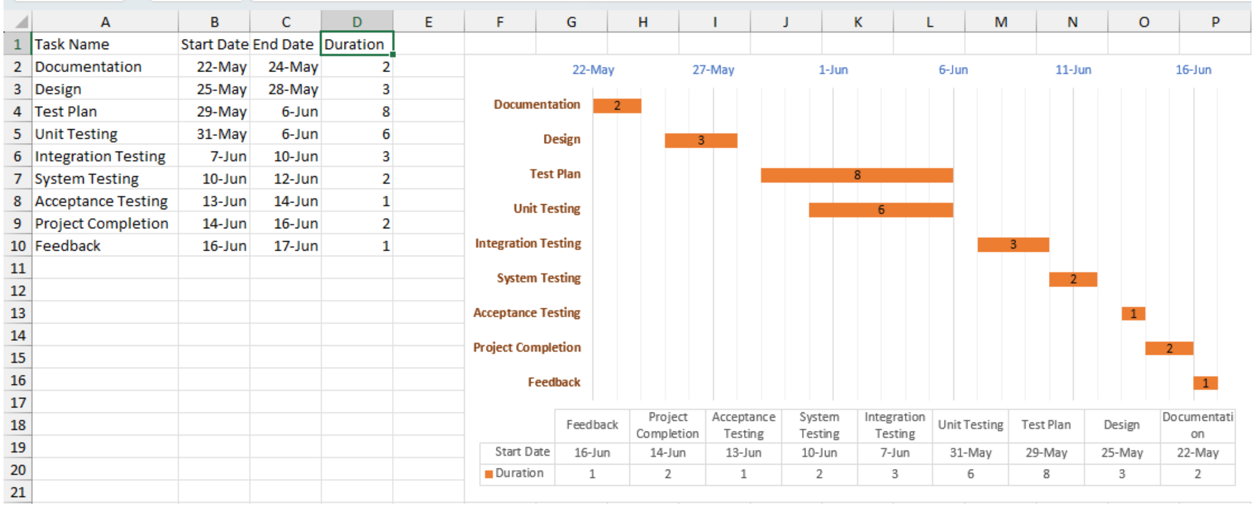
- **System Administrative Training:** They will receive specialized training in server management, system configuration, deployment process and maintenance.

- **Marketing Training:** Training for the marketing team in the system's features and requirements, user interaction, how the platform works so that they can effectively communicate with client about its features and benefits.

11. RESPONSIBILITIES

Task	Test Manager	Project Manager (PM)	Development Team	Test Team	Client
Acceptance Test Documentation & Execution	X	X		X	X
System/Integration Test Documentation & Execution	X	X	X	X	
Unit Test Documentation & Execution	X		X	X	
System Design Review	X	X	X		X
Detail Design Reviews	X	X	X		
Test Procedures & Rules	X		X	X	
Screen & report prototype Reviews	X	X	X	X	X
Change Control & Regression Testing	X	X	X	X	

12. TESTING SCHEDULE



	A	B	C	D
1	Task Name	Start Date	End Date	Duration
2	Documentation	22-May	24-May	2
3	Design	25-May	28-May	3
4	Test Plan	29-May	6-Jun	8
5	Unit Testing	31-May	6-Jun	6
6	Integration Testing	7-Jun	10-Jun	3
7	System Testing	10-Jun	12-Jun	2
8	Acceptance Testing	13-Jun	14-Jun	1
9	Project Completion	14-Jun	16-Jun	2
10	Feedback	16-Jun	17-Jun	1

	Feedback	Project Completion	Acceptance Testing	System Testing	Integration Testing	Unit Testing	Test Plan	Design	Documentation
Start Date	16-Jun	14-Jun	13-Jun	10-Jun	7-Jun	31-May	29-May	25-May	22-May
🟧 Duration	1	2	1	2	3	6	8	3	2

13. PLANNING RISKS AND CONTINGENCIES

S/N	Risk Description	Probability	Impact	Mitigation Plan
1	Unrealistic or short time estimate	40%	Delay project 2 weeks or more overall	Take multiple estimation, adjust project timeline accordingly
2	Development tools & testing environment mismatch	20%	Function running discontinue and inconsistent testing results	Regular check for updates of tools sync & testing between development and testing environment, conduct environment setup reviews
3	Overlooked edge cases in user perspectives	30%	Critical bugs after deployment	Engage in comprehensive thorough testing and use scenarios from user perspectives
4	High number of bugs detected in late testing	25%	Delays in deployment and launch to finish fixing	Frequent regression or retesting after every new feature or unit testing
5	Lacks of resources for critical testing phases	15%	Extended testing cycles	Prioritize and understand task and request additional resources if needed during critical testing phases
6	External system integration failure	10%	Third party service failure or unexpected payment gateway fault	Conduct integration testing early, ensure a backup or contingency for external service failures
7	Limited user feedback during beta testing	35%	Missed or overlooked usability issues	Run extreme beta testing programs and gather feedback significantly
8	Delays and lack of association of client-side approvals	50%	Project timeline extension and delay	Schedule regular meetings to ensure client is on track with approval processes

Table 2: Risk Mitigation Plan for testing