

SOFTWARE PROJECT-2 REPORT

GAppMarket - Android Mobile Apps, Games and eBooks

By

GHANSHYAM VAJA (235530694022)

URL : gappmarketstore.pythonanywhere.com

UNDER GUIDANCE :

Mr. CHIRAG RACHCHH SIR

Ms. DHRUVI SIDHPURA MA'AM

**SUBMITTED To : N.J. SONECHA MANAGEMENT AND
TECHNICAL INSTITUTE - CHANDUVAV**

SUBMITTED To : GUJARAT TECHNOLOGICAL UNIVERSITY



Master of Computer Application(Semester - 3)

Year of Submission : 2024 – 25



NARANDAS JETHALAL SONECHA MANAGEMENT & TECHNICAL INSTITUTE

Approved by AICTE • Affiliated to GTU
At. Chanduvav, Tal. Veraval, Dist. Gir Somnath. (Gujarat)

Certificate

This is to certify that GHANSHYAM BHARATBHAI VAJA
A student of MCA Semester 3 Enrollment No. 235530694022
Has successfully carried out a project entitled GAppMarket towards fulfillment
of the subject SOFTWARE PROJECT - 2(639404)

Date :

Signature of Guide

Seal of the Institute

Signature of HOD

CERTIFICATE of ORIGINALITY

This is to certify that this Project Report entitled "**GAppMarket- Android Mobile Apps, Games and eBooks**" submitted to **Gujarat Technological University** for **Master in computer application - III(MCA - 3)**, is an original work carried out by **Mr. GHANSHYAM VAJA**.

Enrollment Number : 235530694022 under the guidance of **Mr. CHIRAG RACHCHH** and **Ms. DHRUVI SIDHPURA** matter embodied.

this project is a genuine work done by the student and has not been submitted whether to this University or any other university/Institute for the fulfillment of any course of study.

Date :

**Signature of Guide
(CHIRAG RACHCHH SIR)**

Seal of the Institute

Singnature of HOD

**Signature of Guide
(DHRUVI MA'AM))**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

I am grateful to my **PARENTS, COLLEAGUES, PROJECT GUIDE : CHIRAG RACHCHH SIR, Ms. DHRUVI SIDHPURA , HOD : CHIRAG RACHCHH SIR and DIRECTOR : Dr. K.C. DWIWEDI SIR** for the guidance, inspiration and constructive suggestions that helped me in the preparation of this project.

Date :

(GHANSHYAM VAJA)

Project Specifications

TITLE	G App Market - Android Mobile Apps, Games and eBooks
CATEGORY	WEB APPLICATION
FRONTEND TECHNOLOGIES	BOOTSTRAP and JQUERY
BECKEND TECHNOLOGIES	DJANGO
DATABASE	SQLITE3
PROJECT GUIDE	Mr. CHIRAG RACHCHH, Ms. DHRUVI SIDHPURA
SUBMITTED FROM	N.J. SONECHA MANAGEMENT AND TECHNICAL INSTITUTE - CHANDUVAV
SUBMITTED TO	GUJARAT TECHNOLOGICAL UNIVERSITY - AHMEDABAD
SOURCE CODE(GITHUB)	https://github.com/ghanshyamvaja11/GAppMarket/tree/main

Learning during Project Work

1. Understanding and Structuring a Django Project

Working on GAppMarket has significantly enhanced my understanding of structuring a Django project. I learned how to organize models, views, templates, and static files effectively. Creating different apps within the project, such as User, Publisher, and Administrator, taught me the importance of modularity and separation of concerns.

2. Implementing User Authentication and Authorization

Implementing user authentication and authorization was a critical part of the project. I learned how to:

- Use Django's built-in authentication system to manage user login, logout, and registration.
- Create custom user models for different roles (User, Publisher, Administrator).

3. Building Dynamic Web Pages with Django Templates

I gained hands-on experience in using Django templates to create dynamic web pages. This involved:

- Creating reusable templates with `base.html` and extending them for different pages.
- Utilizing template tags and filters to display dynamic content.
- Ensuring consistency in design and layout across various pages.

4. Database Design and ORM Usage

Designing the database schema and working with Django's ORM (Object-Relational Mapping) was another key learning area. I learned how to:

- Define models and set up relationships between them.
- Use migrations to manage database changes.
- Perform CRUD operations using Django's ORM.

5. Handling File Uploads and Media Files

The project required handling file uploads for content like apps, games, and eBooks. I learned to:

- Configure Django to handle media files.
- Create forms for file uploads and validate the uploaded data.
- Serve uploaded files correctly in the application.

6. Front-end Development and UI/UX Design

While focusing primarily on the backend, I also improved my skills in front-end development. This included:

- Using Bootstrap to create a responsive and visually appealing UI.
- Writing custom CSS for specific design requirements.
- Ensuring a consistent look and feel across all pages.

7. Testing and Debugging

I learned the importance of testing and debugging throughout the development process. This involved:

- Writing unit tests for different parts of the application.
- Using Django's built-in test framework.
- Debugging issues using Django's error reporting and logging features.

8. Version Control with Git

Working on a project of this scale underscored the importance of version control. I became proficient in:

- Using Git for version control.
- Managing branches and merging changes.
- Collaborating with others using platforms like GitHub.

9. Deployment and Environment Management

Finally, I learned how to deploy a Django application. This included:

- Setting up a production environment.
- Managing environment variables and configuration settings securely.

Index

TOPIC	PAGE NUMBER
1. Introduction.....	1
1.1. Existing System.....	1
1.2. Need for the New System.....	1
1.3. Objective of the New System.....	1
1.4. Problem Definition.....	2
1.5. Core Components.....	2
1.6. Project Profile.....	3
1.7. Assumptions and Constraints.....	4
1.8. Advantages and Limitations of the Proposed System.....	4
2. Requirement Determination & Analysis.....	5
2.1. Requirement Determination.....	5
2.2. Targeted Users.....	5
3. System Design.....	6
3.1. Use Case Diagram.....	6
3.2. Class Diagram.....	7
3.3. Interaction Diagram.....	9
3.4. Activity Diagram.....	10
3.5. Data Dictionary.....	11
4. Development.....	19
4.1. Coding Standards.....	19
4.2. Screenshots.....	20
5. Agile Documentation.....	48
5.1 Agile Project Charter.....	48
5.2 Agile Roadmap / Schedule.....	49
5.3 Agile Project Plan.....	50
5.4 Agile User Story (Minimum 3 Tasks).....	50
5.5 Agile Release Plan.....	51
5.6 Agile Sprint Backlog.....	52
5.7 Agile Test Plan.....	53
5.8 Earned-value and burn charts.....	54
6. Proposed Enhancements.....	56
7. Conclusion.....	57
8. Bibliography.....	58

1. Introduction

GAppMarket is an innovative digital marketplace designed to revolutionize the way users discover, purchase, and enjoy a diverse range of digital content including apps, games, and eBooks. This platform caters to both consumers seeking quality digital experiences and publishers looking to showcase their latest creations to a global audience. With a focus on user-centric design and robust functionality, GAppMarket aims to provide a seamless and engaging experience for all stakeholders, supported by modern technology and a commitment to excellence in digital content distribution.

1.1. Existing System:

The current landscape of digital content marketplaces is characterized by fragmented platforms that often specialize in specific content types (such as apps or eBooks) and lack integration across different categories. These systems typically offer basic user registration, content browsing, and purchasing functionalities but often fall short in providing a cohesive user experience and comprehensive management tools for publishers. Users may encounter disparate interfaces, inconsistent content quality, and limited options for content discovery and management.

1.2. Need for the New System:

The necessity for a new digital content marketplace arises from several shortcomings observed in existing systems. These include:

- **Fragmentation:** Users often need to navigate multiple platforms to access different types of digital content, leading to a disjointed user experience.
- **Management Complexity:** Publishers face challenges in managing and distributing content across various platforms, each with its own set of requirements and limitations.
- **Lack of Analytics:** Existing systems may provide basic sales data but lack detailed analytics and insights into user behavior, content performance, and market trends.
- **Security and Scalability:** With the increasing volume of digital content and user interactions, there is a growing need for robust security measures and scalable infrastructure to support a large user base and content catalog.

1.3. Objective of the New System:

The primary objective of the GAppMarket project is to develop a unified, robust digital content marketplace that addresses the aforementioned challenges. Key objectives include:

- **Enhanced User Experience:** Provide a seamless and intuitive interface for users to discover, purchase, and manage digital content across various categories (such as apps, games, and eBooks).
- **Empower Publishers:** Offer publishers streamlined tools for uploading, managing, and monetizing their digital content, including detailed analytics and performance metrics.
- **Administrative Oversight:** Equip administrators with comprehensive controls for managing users, publishers, content categories, system settings and more to ensure smooth operation and compliance.
- **Scalability and Security:** Implement scalable architecture and robust security measures to handle increasing user traffic, content uploads, and financial transactions securely.

1.4. Problem Definition:

The GAppMarket project aims to address the following key challenges:

- **User Fragmentation:** Eliminate the need for users to visit multiple platforms by consolidating various types of digital content into a single marketplace.
- **Publishing Complexity:** Simplify the content management process for publishers, enabling them to efficiently upload, categorize, and update their digital offerings.
- **Insufficient Analytics:** Provide detailed insights into user engagement, content performance, and sales trends to empower publishers and administrators in making data-driven decisions.
- **Security and Trust:** Ensure secure transactions, data privacy, and content authenticity to build trust among users, publishers, and stakeholders.

1.5. Core Components:

The core components of the GAppMarket system include:

- **User Management:** Authentication, registration, and profile management functionalities for users.
- **Content Management:** Tools for publishers to upload, categorize, update, and withdraw digital content (such as apps, games, and eBooks).
- **Search and Discovery:** Advanced search capabilities and personalized recommendations to enhance content discoverability for users.
- **Analytics and Reporting:** Dashboards and reports for publishers and administrators to monitor content performance, user engagement, sales metrics, and trends.
- **Administrative Controls:** Comprehensive tools for administrators to manage user accounts, publisher relationships, content categories, and platform settings.
- **UI/UX Design:** Responsive and user-friendly interface design that ensures a consistent and intuitive experience across desktop and mobile devices.
- **Technologies:**

1. Django:

- **URL Routing:** Maps URL patterns to views.
- **Models:** Defines data models and relationships using ORM (Object-Relational Mapping).
- **Views:** Processes user requests and returns responses.
- **Templates:** Renders HTML pages dynamically using Django's template engine.
- **Admin Interface:** Automatic admin interface for managing application data.

2. Bootstrap:

- **Grid System:** Responsive layout grid system for arranging content.
- **UI Components:** Pre-styled buttons, forms, navigation bars, and other elements.
- **Utilities:** Helper classes for spacing, alignment, and typography.
- **Customization:** Allows customization through SASS variables and mixins.
- **Responsive Design:** Ensures consistent appearance across different devices and screen sizes.

3. jQuery:

- **DOM Manipulation:** Selects and manipulates HTML elements.
- **Event Handling:** Binds actions to user interactions (e.g., clicks, keystrokes).
- **AJAX:** Simplifies asynchronous data retrieval and updates.
- **Animations:** Adds visual effects and animations to web pages.
- **Plugins:** Extends functionality with numerous plugins for various tasks.

4. SQLite3:

- **Relational Database:** Stores structured data in tables with defined relationships.
- **SQL Queries:** Executes SQL queries for data retrieval, insertion, updating, and deletion.
- **ACID Compliance:** Ensures data integrity with Atomicity, Consistency, Isolation, and Durability.
- **Transactions:** Groups multiple operations into a single unit of work.
- **Indexes:** Optimizes data retrieval with indexes on columns for faster queries.

1.6. Project Profile:

- **Name:** GAppMarket
- **Type:** Digital Content Marketplace
- **Scope:** Initially focused on apps(Android), games(Android), and eBooks with potential expansion to include other digital content types (such as music, videos, and educational resources).
- **Technologies:** Developed using **Django framework** for backend development, **Bootstrap framework** for frontend design, **sqlite3 database** for data storage, and hosted on **pythonanywhere**.
- **Developed By :** GHANSHYAM VAJA
- **Purpose :** MCA Semester 3 Project Submission

1.7. Assumptions and Constraints:

- **Assumptions**
 - **Internet Connectivity:** Users are assumed to have access to reliable internet connectivity, which is necessary for browsing, purchasing, and interacting with digital content on the platform.
 - **Web Browser Compatibility:** Users are expected to use modern web browsers capable of supporting interactive web applications, ensuring a seamless and responsive user experience.
 - **Publisher Skills:** Publishers are assumed to possess the technical skills required to upload, manage, and maintain digital content in the formats specified by the platform. This includes understanding file formats, quality standards, and metadata requirements.

- **Constraints**

- **Budget Limitations:** The project operates under initial budget constraints, which may restrict expenditures on marketing campaigns and promotional activities aimed at attracting and retaining users.
- **Scalability Challenges:** Potential challenges may arise in scaling the platform's infrastructure to accommodate a growing user base and increasing content catalog. This includes managing server load, optimizing performance, and ensuring system stability during periods of rapid growth.
- **Content Quality:** Ensuring consistent high-quality content from publishers is crucial. Maintaining standards for accuracy, legality, and user experience is essential for user satisfaction and platform credibility

1.8. Advantages and Limitations of the Proposed System:

- **Advantages**

- **Unified Platform:** Consolidates various types of digital content into a single marketplace, offering users a seamless browsing and purchasing experience.
- **Empowered Publishers:** Provides publishers with powerful tools for content management, analytics, and monetization, thereby enhancing their ability to reach and engage with a broader audience.
- **Comprehensive Analytics:** Offers detailed insights into user behavior, content performance, and market trends, enabling informed decision-making for publishers and administrators.
- **Scalability and Security:** Built on scalable architecture with robust security measures to support growth in user base and content volume while safeguarding transactions and user data.

- **Limitations**

- **Initial Scalability Challenges:** May experience initial challenges in scaling infrastructure and managing increased server load during periods of rapid growth.
- **Dependency on Content Quality:** Success hinges on the quality and relevance of digital content uploaded by publishers. Poor-quality content may affect user satisfaction and retention.
- **Security Risks:** Potential vulnerabilities in data privacy and content piracy require continuous monitoring and updates to maintain trust and compliance with legal and regulatory standards.

2. Requirement Determination & Analysis

2.1. Requirement Determination:

Requirement determination involves identifying and documenting the functional and non-functional requirements of the GAppMarket project. This process includes:

- **Functional Requirements:** Specifications of system capabilities, such as user registration, content upload, search functionality, transaction processing, and administrative controls.
- **Non-functional Requirements:** Criteria that specify how the system should behave, such as performance metrics (response time, scalability), security measures (data encryption, access control), usability (user interface design, accessibility), and reliability (system uptime, error handling).

2.2. Targeted Users:

The GAppMarket project targets the following user groups:

- **End Users:** Individuals interested in purchasing and consuming digital content, including apps, games, and eBooks. They interact with the platform to browse, search, purchase, and manage their digital library.
- **Publishers:** Content creators and distributors who upload and manage digital content on the platform. They use tools provided by the system to upload content, set pricing, monitor performance metrics, and engage with user feedback.
- **Administrators:** System administrators responsible for overseeing the entire platform. They manage user accounts, monitor content quality and compliance, configure system settings, etc.

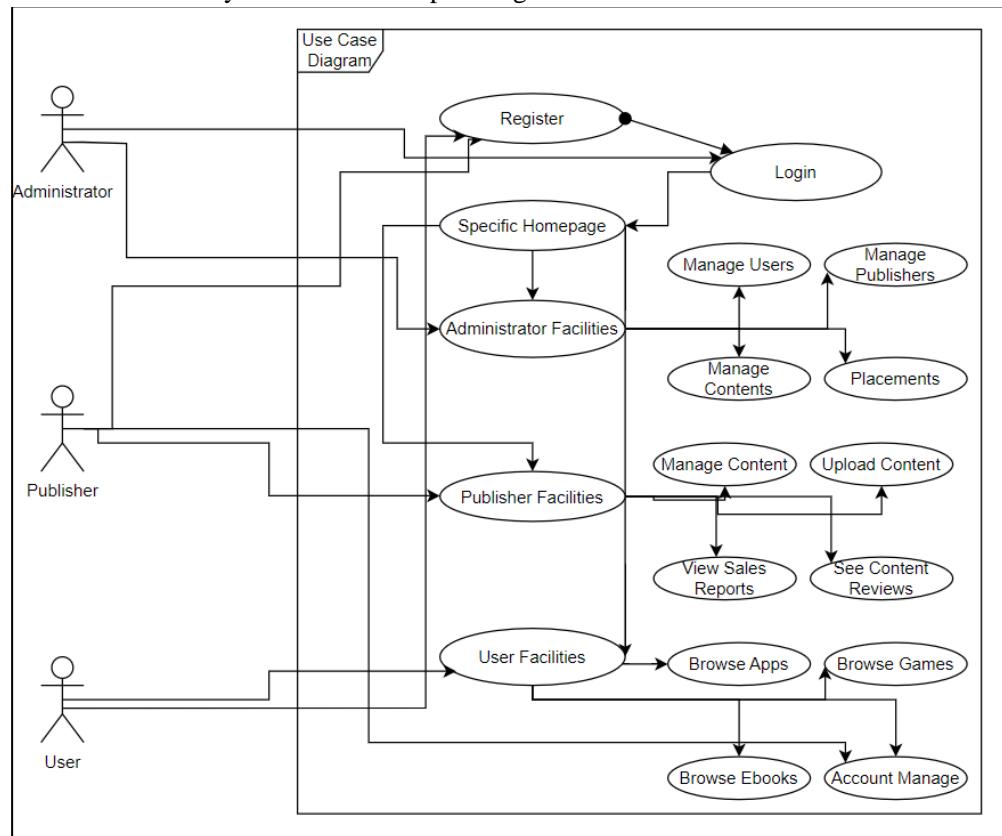
3. System Design

In the blueprint of our GAppMarket, the system design plays a pivotal role. This section unfolds the design aspects through various diagrams, encapsulating the system's architecture and functionality.

3.1 Use Case Diagram

Requirement determination involves identifying and documenting the functional and non-functional requirements of the GAppMarket project. This process includes:

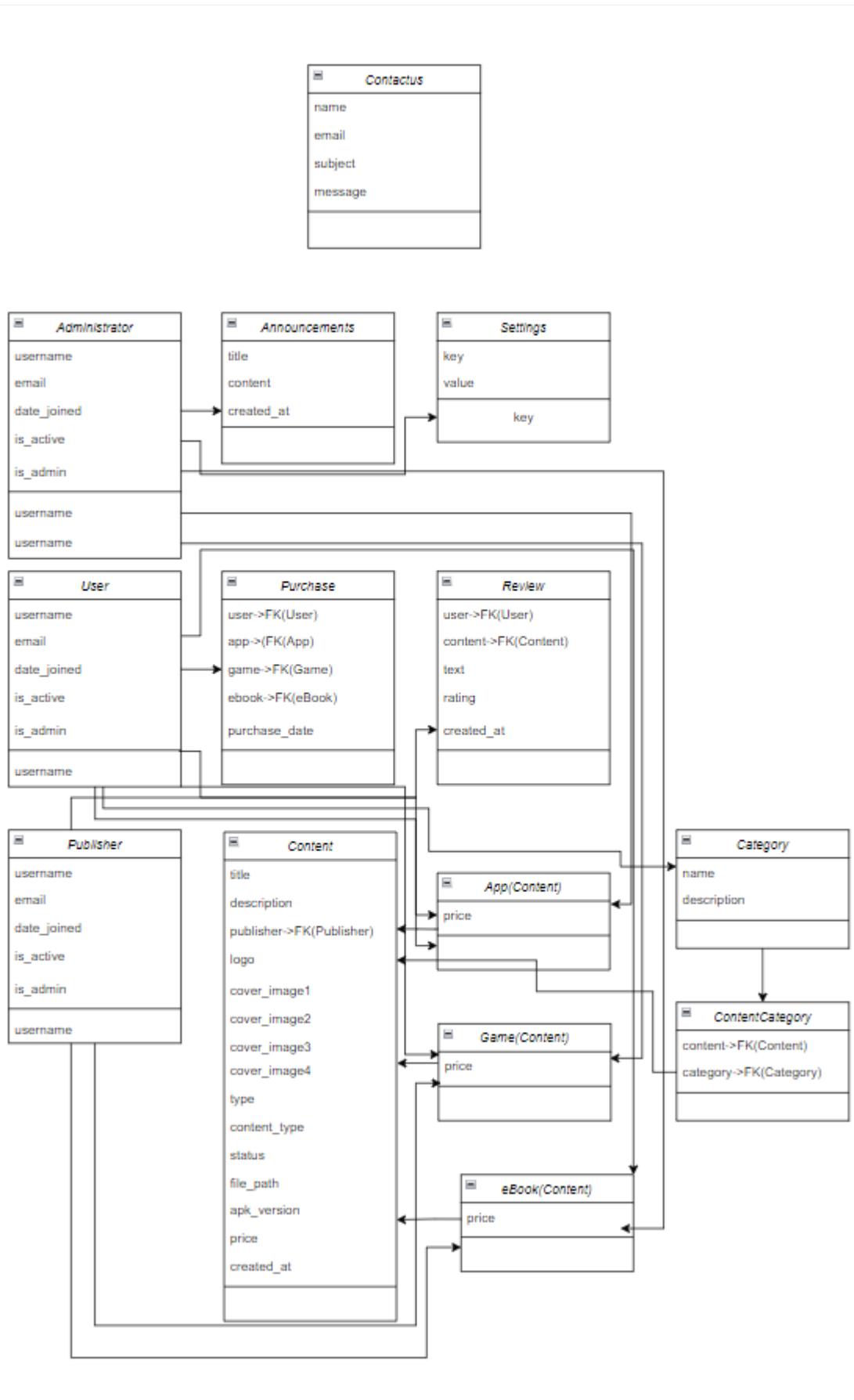
- **Actors:** End Users, Publishers, Administrators.
- **Use Cases:** Actions and functionalities each actor can perform within the system, such as "Browse Content," "Purchase Content," "Upload Content," "Manage User Accounts," "Generate Sales Reports," etc.
- **Relationships:** Associations between actors and use cases, illustrating how each actor interacts with the system to achieve specific goals.



3.2. Class Diagram:

The Class Diagram for the GAppMarket project illustrates the static structure of the system by showing:

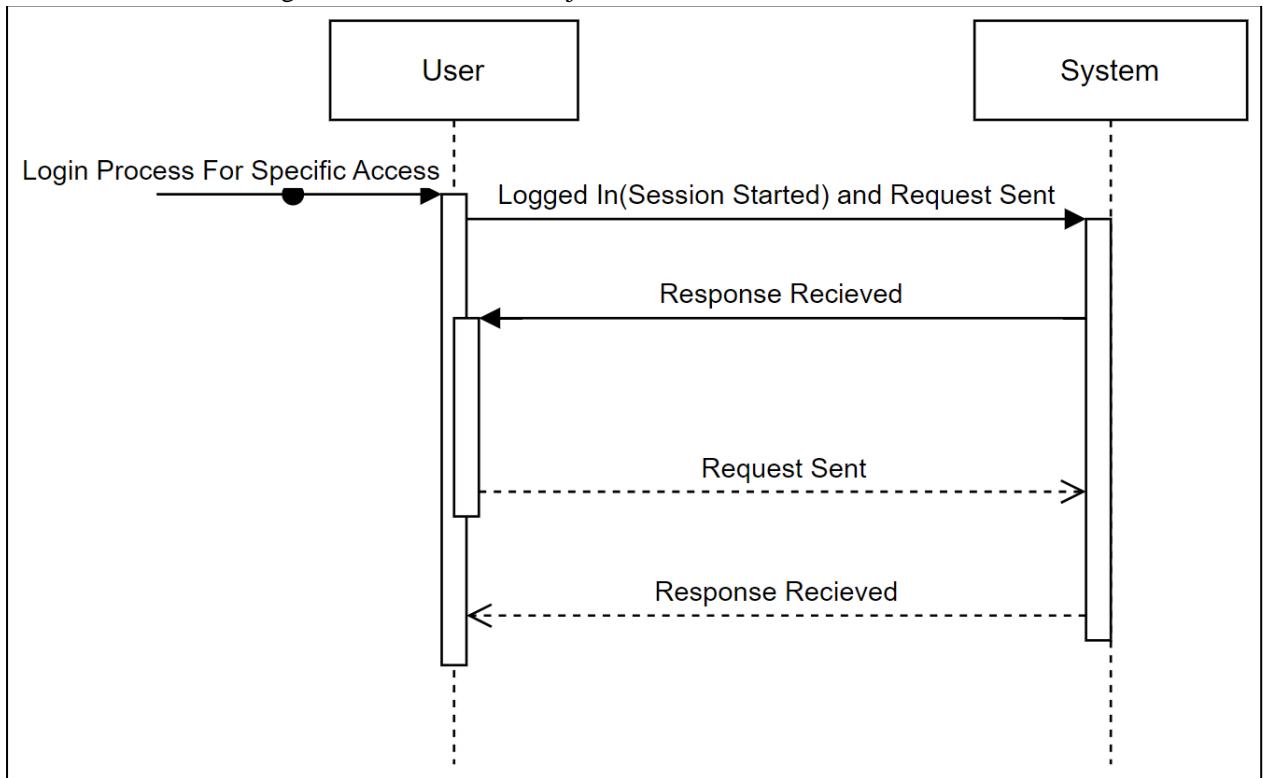
- **Classes:** Representing entities such as User, Publisher, Content (App, Game, eBook), Category, Order etc.
- **Attributes:** Properties or data fields of each class, such as username, email for User; title, description for Content; name, description for Category, etc.
- **Relationships:** Associations between classes, including:
 - **Association:** Connections between classes indicating how instances are related (e.g., User purchases Order, Publisher uploads Content).
 - **Inheritance:** Hierarchical relationships showing inheritance (e.g., Publisher inherits from User).
 - **Composition/Aggregation:** Whole-part relationships (e.g., ContentCategory aggregates Content and Category).



3.3. Interaction Diagram:

The Interaction Diagram delves into the dynamic aspects of the system, illustrating the flow of interactions between different components. Through sequence diagrams or collaboration diagrams, it showcases how users, entities, and system modules interact during various processes, providing insights into the system's runtime behavior.

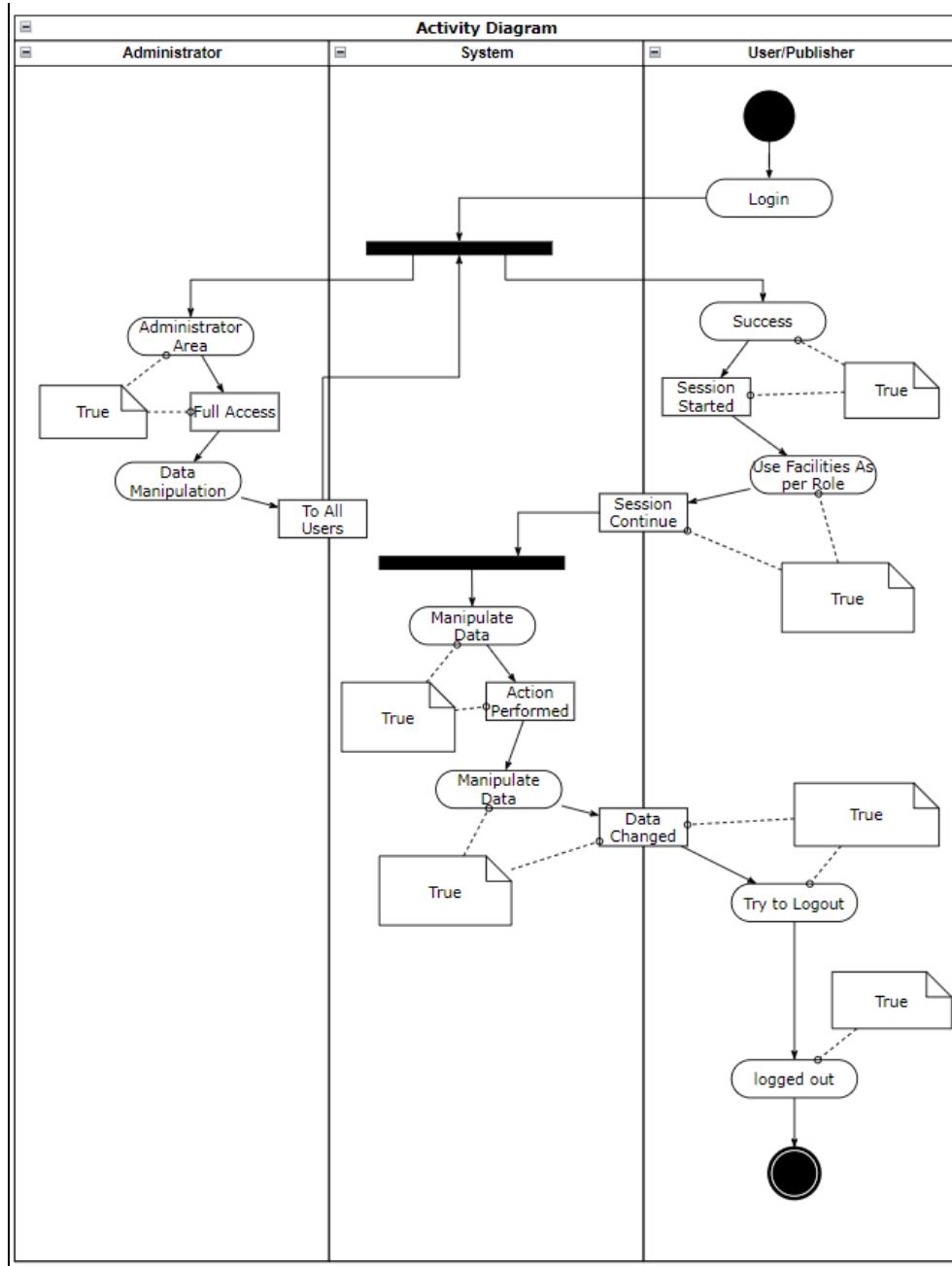
The Sequence Diagram for GAppMarket illustrates the dynamic behavior of the system by showing interactions between objects over time.



3.4. Activity Diagram:

The Activity Diagram for GAppMarket depicts workflows or processes within the system, focusing on:

- **Activities:** Actions or steps in a process, such as "User Registration," "Content Upload," "Order Processing."
- **Decisions:** Points where the flow can diverge based on conditions, such as user choices or system responses.
- **Transitions:** Arrows indicating the flow of control from one activity to another.



3.5. Data Dictionary:

The Activity Diagram for GAppMarket depicts workflows or processes within the system, focusing on:

- **Entities:** Data entities like User, Publisher, Content, Category, Order, etc.
- **Attributes:** Characteristics or properties of each entity, such as username, email for User; title, description for Content.
- **Data Types:** Specification of data types for attributes, such as string, integer, date.
- **Constraints:** Rules or conditions that apply to data, such as uniqueness constraints for usernames, format constraints for dates.

Tables (25) :

Name	Type	Schema
Administrator_administrator		CREATE TABLE "Administrator_administrator" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "email" varchar(255) NOT NULL UNIQUE, "password" varchar(128) NOT NULL, "date_joined" datetime NOT NULL, "is_active" bool NOT NULL, "is_admin" bool NOT NULL, "last_login" datetime NULL, "username" varchar(100) NOT NULL UNIQUE)
id	integer	"id" integer NOT NULL
email	varchar(255)	"email" varchar(255) NOT NULL UNIQUE
password	varchar(128)	"password" varchar(128) NOT NULL
date_joined	datetime	"date_joined" datetime NOT NULL
is_active	bool	"is_active" bool NOT NULL
is_admin	bool	"is_admin" bool NOT NULL
last_login	datetime	"last_login" datetime
username	varchar(100)	"username" varchar(100) NOT NULL UNIQUE
Administrator_announcement		CREATE TABLE "Administrator_announcement" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title" varchar(200) NOT NULL, "content" text NOT NULL, "created_at" datetime NOT NULL)
id	integer	"id" integer NOT NULL
title	varchar(200)	"title" varchar(200) NOT NULL
content	text	"content" text NOT NULL
created_at	datetime	"created_at" datetime NOT NULL
Administrator_settings		CREATE TABLE "Administrator_settings" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "key" varchar(100) NOT NULL UNIQUE, "value" varchar(255) NOT NULL)
id	integer	"id" integer NOT NULL
key	varchar(100)	"key" varchar(100) NOT NULL UNIQUE
value	varchar(255)	"value" varchar(255) NOT NULL
Home_contactus		CREATE TABLE "Home_contactus" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(100) NOT NULL, "email" varchar(254) NOT NULL, "subject" varchar(200) NOT NULL, "message" text NOT NULL)
id	integer	"id" integer NOT NULL
name	varchar(100)	"name" varchar(100) NOT NULL
email	varchar(254)	"email" varchar(254) NOT NULL
subject	varchar(200)	"subject" varchar(200) NOT NULL
message	text	"message" text NOT NULL
Publisher_app		CREATE TABLE "Publisher_app" ("content_ptr_id" bigint NOT NULL PRIMARY KEY REFERENCES "Publisher_content" ("id") DEFERRABLE INITIALLY DEFERRED, "price" decimal NULL)
content_ptr_id	bigint	"content_ptr_id" bigint NOT NULL
price	decimal	"price" decimal
Publisher_category		CREATE TABLE "Publisher_category" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(100) NOT NULL UNIQUE, "description" text NOT NULL)
id	integer	"id" integer NOT NULL
name	varchar(100)	"name" varchar(100) NOT NULL UNIQUE
description	text	"description" text NOT NULL

Name	Type	Schema
Publisher_content		<pre>CREATE TABLE "Publisher_content" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title" varchar(200) NOT NULL, "description" text NOT NULL, "logo" varchar(100) NOT NULL, "cover_image1" varchar(100) NOT NULL, "cover_image2" varchar(100) NOT NULL, "cover_image3" varchar(100) NOT NULL, "cover_image4" varchar(100) NOT NULL, "type" varchar(4) NOT NULL, "status" varchar(20) NOT NULL, "publisher_id" bigint NOT NULL REFERENCES "Publisher_publisher" ("id") DEFERRABLE INITIALLY DEFERRED, "file_path" varchar(150) NOT NULL, "apk_version" varchar(20) NOT NULL, "content_type" varchar(5) NOT NULL, "created_at" datetime NOT NULL)</pre>
id	integer	"id" integer NOT NULL
title	varchar(200)	"title" varchar(200) NOT NULL
description	text	"description" text NOT NULL
logo	varchar(100)	"logo" varchar(100) NOT NULL
cover_image1	varchar(100)	"cover_image1" varchar(100) NOT NULL
cover_image2	varchar(100)	"cover_image2" varchar(100) NOT NULL
cover_image3	varchar(100)	"cover_image3" varchar(100) NOT NULL
cover_image4	varchar(100)	"cover_image4" varchar(100) NOT NULL
type	varchar(4)	"type" varchar(4) NOT NULL
status	varchar(20)	"status" varchar(20) NOT NULL
publisher_id	bigint	"publisher_id" bigint NOT NULL
file_path	varchar(150)	"file_path" varchar(150) NOT NULL
apk_version	varchar(20)	"apk_version" varchar(20) NOT NULL
content_type	varchar(5)	"content_type" varchar(5) NOT NULL
created_at	datetime	"created_at" datetime NOT NULL
Publisher_contentcategory		<pre>CREATE TABLE "Publisher_contentcategory" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "category_id" bigint NOT NULL REFERENCES "Publisher_category" ("id") DEFERRABLE INITIALLY DEFERRED, "content_id" bigint NOT NULL REFERENCES "Publisher_content" ("id") DEFERRABLE INITIALLY DEFERRED)</pre>
id	integer	"id" integer NOT NULL
category_id	bigint	"category_id" bigint NOT NULL
content_id	bigint	"content_id" bigint NOT NULL
Publisher_ebook		<pre>CREATE TABLE "Publisher_ebook" ("content_ptr_id" bigint NOT NULL PRIMARY KEY REFERENCES "Publisher_content" ("id") DEFERRABLE INITIALLY DEFERRED, "price" decimal NULL)</pre>
content_ptr_id	bigint	"content_ptr_id" bigint NOT NULL
price	decimal	"price" decimal
Publisher_game		<pre>CREATE TABLE "Publisher_game" ("content_ptr_id" bigint NOT NULL PRIMARY KEY REFERENCES "Publisher_content" ("id") DEFERRABLE INITIALLY DEFERRED, "price" decimal NULL)</pre>
content_ptr_id	bigint	"content_ptr_id" bigint NOT NULL
price	decimal	"price" decimal
Publisher_publisher		<pre>CREATE TABLE "Publisher_publisher" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "email" varchar(255) NOT NULL UNIQUE, "password" varchar(128) NOT NULL, "date_joined" datetime NOT NULL, "is_active" bool NOT NULL, "is_admin" bool NOT NULL)</pre>

Name	Type	Schema
		bool NOT NULL, "last_login" datetime NULL, "username" varchar(100) NOT NULL UNIQUE)
id	integer	"id" integer NOT NULL
email	varchar(255)	"email" varchar(255) NOT NULL UNIQUE
password	varchar(128)	"password" varchar(128) NOT NULL
date_joined	datetime	"date_joined" datetime NOT NULL
is_active	bool	"is_active" bool NOT NULL
is_admin	bool	"is_admin" bool NOT NULL
last_login	datetime	"last_login" datetime
username	varchar(100)	"username" varchar(100) NOT NULL UNIQUE
User_purchase		CREATE TABLE "User_purchase" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "purchase_date" datetime NOT NULL, "user_id" bigint NOT NULL REFERENCES "User_user" ("id") DEFERRABLE INITIALLY DEFERRED, "ebook_id" bigint NULL REFERENCES "Publisher_ebook" ("content_ptr_id") DEFERRABLE INITIALLY DEFERRED, "game_id" bigint NULL REFERENCES "Publisher_game" ("content_ptr_id") DEFERRABLE INITIALLY DEFERRED, "app_id" bigint NULL REFERENCES "Publisher_app" ("content_ptr_id") DEFERRABLE INITIALLY DEFERRED)
id	integer	"id" integer NOT NULL
purchase_date	datetime	"purchase_date" datetime NOT NULL
user_id	bigint	"user_id" bigint NOT NULL
ebook_id	bigint	"ebook_id" bigint
game_id	bigint	"game_id" bigint
app_id	bigint	"app_id" bigint
User_review		CREATE TABLE "User_review" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "text" text NOT NULL, "created_at" datetime NOT NULL, "content_id" bigint NOT NULL REFERENCES "Publisher_content" ("id") DEFERRABLE INITIALLY DEFERRED, "user_id" bigint NOT NULL REFERENCES "User_user" ("id") DEFERRABLE INITIALLY DEFERRED, "rating" integer NOT NULL)
id	integer	"id" integer NOT NULL
text	text	"text" text NOT NULL
created_at	datetime	"created_at" datetime NOT NULL
content_id	bigint	"content_id" bigint NOT NULL
user_id	bigint	"user_id" bigint NOT NULL
rating	integer	"rating" integer NOT NULL
User_user		CREATE TABLE "User_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "email" varchar(255) NOT NULL UNIQUE, "password" varchar(128) NOT NULL, "date_joined" datetime NOT NULL, "is_active" bool NOT NULL, "is_admin" bool NOT NULL, "last_login" datetime NULL, "username" varchar(100) NOT NULL UNIQUE)
id	integer	"id" integer NOT NULL
email	varchar(255)	"email" varchar(255) NOT NULL UNIQUE
password	varchar(128)	"password" varchar(128) NOT NULL
date_joined	datetime	"date_joined" datetime NOT NULL
is_active	bool	"is_active" bool NOT NULL
is_admin	bool	"is_admin" bool NOT NULL
last_login	datetime	"last_login" datetime

Name	Type	Schema
username	varchar(100)	"username" varchar(100) NOT NULL UNIQUE
auth_group		CREATE TABLE "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE)
id	integer	"id" integer NOT NULL
name	varchar(150)	"name" varchar(150) NOT NULL UNIQUE
auth_group_permissions		CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED)
id	integer	"id" integer NOT NULL
group_id	integer	"group_id" integer NOT NULL
permission_id	integer	"permission_id" integer NOT NULL
auth_permission		CREATE TABLE "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "codename" varchar(100) NOT NULL, "name" varchar(255) NOT NULL)
id	integer	"id" integer NOT NULL
content_type_id	integer	"content_type_id" integer NOT NULL
codename	varchar(100)	"codename" varchar(100) NOT NULL
name	varchar(255)	"name" varchar(255) NOT NULL
auth_user		CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NULL, "is_superuser" bool NOT NULL, "username" varchar(150) NOT NULL UNIQUE, "last_name" varchar(150) NOT NULL, "email" varchar(254) NOT NULL, "is_staff" bool NOT NULL, "is_active" bool NOT NULL, "date_joined" datetime NOT NULL, "first_name" varchar(150) NOT NULL)
id	integer	"id" integer NOT NULL
password	varchar(128)	"password" varchar(128) NOT NULL
last_login	datetime	"last_login" datetime
is_superuser	bool	"is_superuser" bool NOT NULL
username	varchar(150)	"username" varchar(150) NOT NULL UNIQUE
last_name	varchar(150)	"last_name" varchar(150) NOT NULL
email	varchar(254)	"email" varchar(254) NOT NULL
is_staff	bool	"is_staff" bool NOT NULL
is_active	bool	"is_active" bool NOT NULL
date_joined	datetime	"date_joined" datetime NOT NULL
first_name	varchar(150)	"first_name" varchar(150) NOT NULL
auth_user_groups		CREATE TABLE "auth_user_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED)
id	integer	"id" integer NOT NULL
user_id	integer	"user_id" integer NOT NULL
group_id	integer	"group_id" integer NOT NULL
auth_user_user_permissions		CREATE TABLE "auth_user_user_permissions" ("id"

Indices (25) :

Name	Type	Schema
		integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED)
id	integer	"id" integer NOT NULL
user_id	integer	"user_id" integer NOT NULL
permission_id	integer	"permission_id" integer NOT NULL
django_admin_log		CREATE TABLE "django_admin_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "object_id" text NULL, "object_repr" varchar(200) NOT NULL, "action_flag" smallint unsigned NOT NULL CHECK ("action_flag" >= 0), "change_message" text NOT NULL, "content_type_id" integer NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "action_time" datetime NOT NULL)
id	integer	"id" integer NOT NULL
object_id	text	"object_id" text
object_repr	varchar(200)	"object_repr" varchar(200) NOT NULL
action_flag	smallint unsigned	"action_flag" smallint unsigned NOT NULL CHECK("action_flag" >= 0)
change_message	text	"change_message" text NOT NULL
content_type_id	integer	"content_type_id" integer
user_id	integer	"user_id" integer NOT NULL
action_time	datetime	"action_time" datetime NOT NULL
django_content_type		CREATE TABLE "django_content_type" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app_label" varchar(100) NOT NULL, "model" varchar(100) NOT NULL)
id	integer	"id" integer NOT NULL
app_label	varchar(100)	"app_label" varchar(100) NOT NULL
model	varchar(100)	"model" varchar(100) NOT NULL
django_migrations		CREATE TABLE "django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "applied" datetime NOT NULL)
id	integer	"id" integer NOT NULL
app	varchar(255)	"app" varchar(255) NOT NULL
name	varchar(255)	"name" varchar(255) NOT NULL
applied	datetime	"applied" datetime NOT NULL
django_session		CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datetime NOT NULL)
session_key	varchar(40)	"session_key" varchar(40) NOT NULL
session_data	text	"session_data" text NOT NULL
expire_date	datetime	"expire_date" datetime NOT NULL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"

Name	Type	Schema
Publisher_content_publisher_id_5458a617		CREATE INDEX "Publisher_content_publisher_id_5458a617" ON "Publisher_content" ("publisher_id")
publisher_id		"publisher_id"
Publisher_contentcategory_id_24843e41		CREATE INDEX "Publisher_contentcategory_id_24843e41" ON "Publisher_contentcategory" ("category_id")
category_id		"category_id"
Publisher_contentcategory_content_id_293a2973		CREATE INDEX "Publisher_contentcategory_content_id_293a2973" ON "Publisher_contentcategory" ("content_id")
content_id		"content_id"
Publisher_contentcategory_content_id_category_id_93222e12_uniq		CREATE UNIQUE INDEX "Publisher_contentcategory_content_id_category_id_93222e12_uniq" ON "Publisher_contentcategory" ("content_id", "category_id")
content_id		"content_id"
category_id		"category_id"
User_purchase_app_id_d59663b4		CREATE INDEX "User_purchase_app_id_d59663b4" ON "User_purchase" ("app_id")
app_id		"app_id"
User_purchase_ebook_id_dbd841c0		CREATE INDEX "User_purchase_ebook_id_dbd841c0" ON "User_purchase" ("ebook_id")
ebook_id		"ebook_id"
User_purchase_game_id_11ebf768		CREATE INDEX "User_purchase_game_id_11ebf768" ON "User_purchase" ("game_id")
game_id		"game_id"
User_purchase_user_id_bbf3888e		CREATE INDEX "User_purchase_user_id_bbf3888e" ON "User_purchase" ("user_id")
user_id		"user_id"
User_review_content_id_76a563e1		CREATE INDEX "User_review_content_id_76a563e1" ON "User_review" ("content_id")
content_id		"content_id"
User_review_user_id_e9645489		CREATE INDEX "User_review_user_id_e9645489" ON "User_review" ("user_id")
user_id		"user_id"
auth_group_permissions_group_id_b120cbf9		CREATE INDEX "auth_group_permissions_group_id_b120cbf9" ON "auth_group_permissions" ("group_id")
group_id		"group_id"
auth_group_permissions_group_id_permission_id_0cd325b0_uniq		CREATE UNIQUE INDEX "auth_group_permissions_group_id_permission_id_0cd325b0_uniq" ON "auth_group_permissions" ("group_id", "permission_id")
group_id		"group_id"
permission_id		"permission_id"
auth_group_permissions_permission_id_84c5c92e		CREATE INDEX "auth_group_permissions_permission_id_84c5c92e" ON "auth_group_permissions" ("permission_id")
permission_id		"permission_id"

Name	Type	Schema
auth_permission_content_type_id_2f476e4b		CREATE INDEX "auth_permission_content_type_id_2f476e4b" ON "auth_permission" ("content_type_id")
content_type_id		"content_type_id"
auth_permission_content_type_id_codename_01ab375a_uniq		CREATE UNIQUE INDEX "auth_permission_content_type_id_codename_01ab375a_uniq" ON "auth_permission" ("content_type_id", "codename")
content_type_id		"content_type_id"
codename		"codename"
auth_user_groups_group_id_97559544		CREATE INDEX "auth_user_groups_group_id_97559544" ON "auth_user_groups" ("group_id")
group_id		"group_id"
auth_user_groups_user_id_6a12ed8b		CREATE INDEX "auth_user_groups_user_id_6a12ed8b" ON "auth_user_groups" ("user_id")
user_id		"user_id"
auth_user_groups_user_id_group_id_94350c0c_uniq		CREATE UNIQUE INDEX "auth_user_groups_user_id_group_id_94350c0c_uniq" ON "auth_user_groups" ("user_id", "group_id")
user_id		"user_id"
group_id		"group_id"
auth_user_user_permissions_permission_id_1fb5f2c2c		CREATE INDEX "auth_user_user_permissions_permission_id_1fb5f2c" ON "auth_user_user_permissions" ("permission_id")
permission_id		"permission_id"
auth_user_user_permissions_user_id_a95ead1b		CREATE INDEX "auth_user_user_permissions_user_id_a95ead1b" ON "auth_user_user_permissions" ("user_id")
user_id		"user_id"
auth_user_user_permissions_user_id_permission_id_14a6b632_uniq		CREATE UNIQUE INDEX "auth_user_user_permissions_user_id_permission_id_14a6b632_uniq" ON "auth_user_user_permissions" ("user_id", "permission_id")
user_id		"user_id"
permission_id		"permission_id"
django_admin_log_content_type_id_c4bce8eb		CREATE INDEX "django_admin_log_content_type_id_c4bce8eb" ON "django_admin_log" ("content_type_id")
content_type_id		"content_type_id"
django_admin_log_user_id_c564eba6		CREATE INDEX "django_admin_log_user_id_c564eba6" ON "django_admin_log" ("user_id")
user_id		"user_id"
django_content_type_app_label_model_76bd3d3b_uniq		CREATE UNIQUE INDEX "django_content_type_app_label_model_76bd3d3b_uniq" ON "django_content_type" ("app_label", "model")
app_label		"app_label"
model		"model"
django_session_expire_d		CREATE INDEX "django_session_expire_date_a5c62663" ON "django_session" ("expire_date")
expire_date		"expire_date"

4. Development

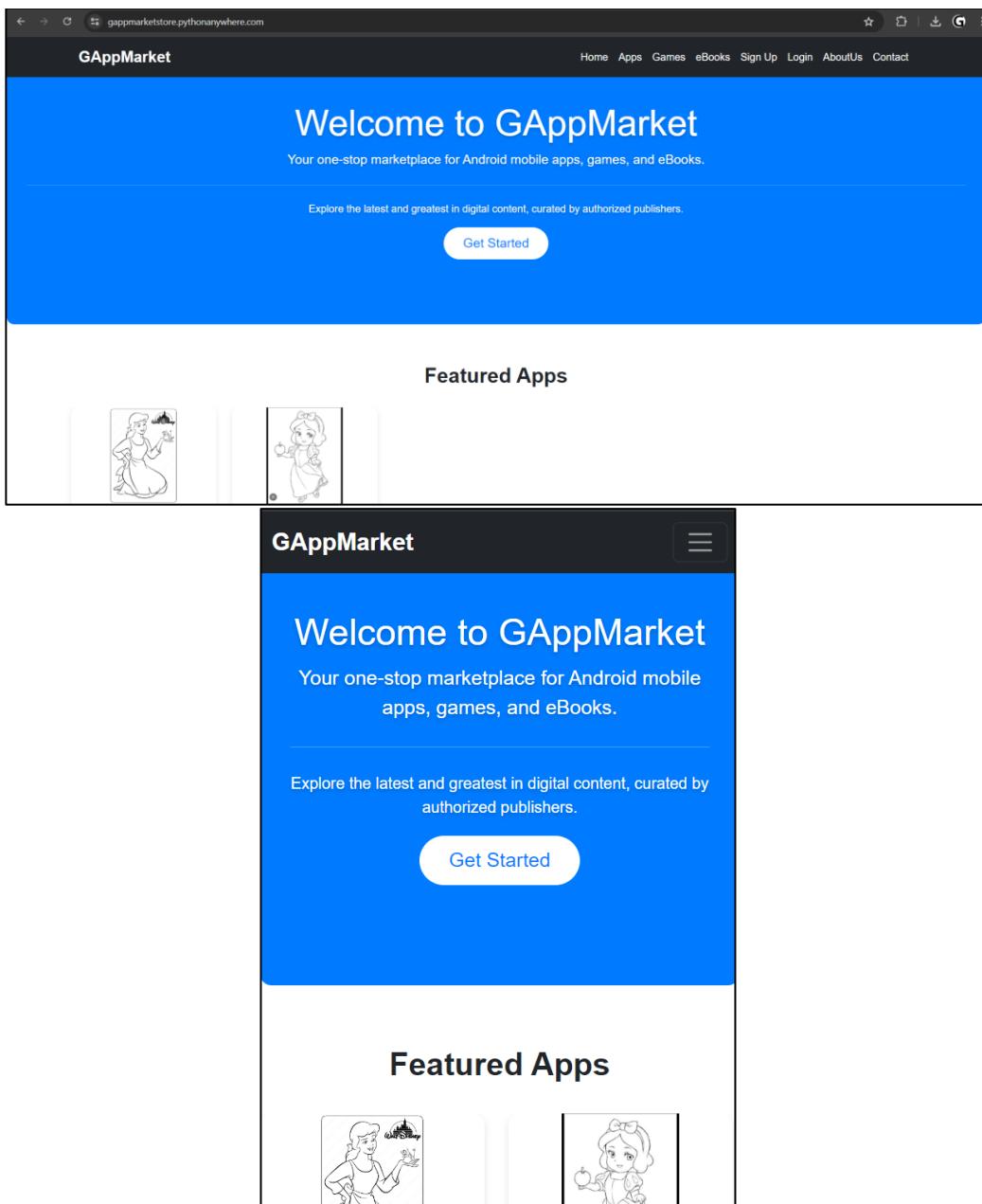
4.1 Coding Standards

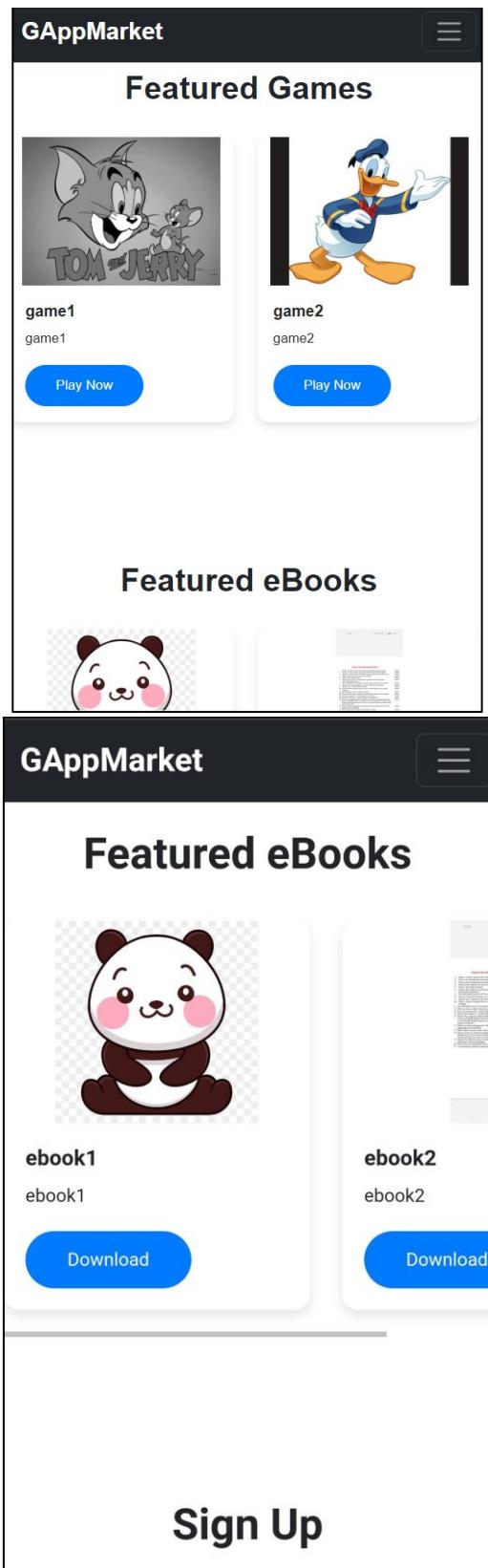
Coding standards are guidelines that define how code should be written and formatted within the GAppMarket project. They ensure consistency, readability, and maintainability across the codebase. Here are key aspects of coding standards for the project:

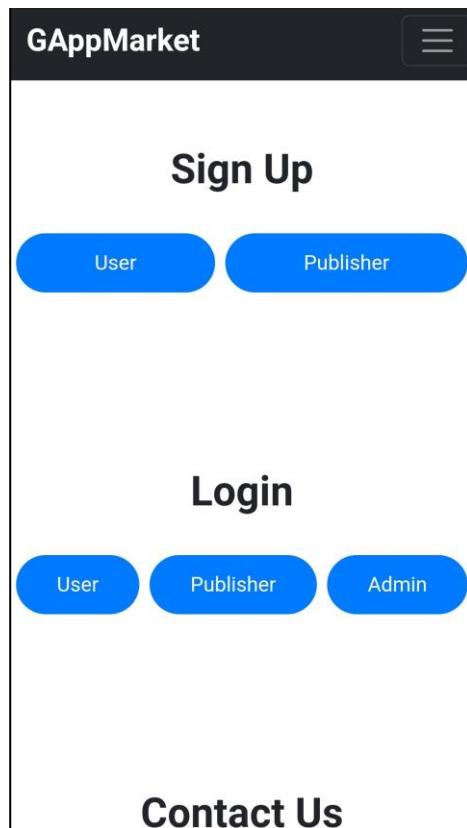
1. **Naming Conventions:**
 - o **Variables and Functions:** Use meaningful and descriptive names. Follow camelCase or _ (for separation) for variables and functions (e.g., `userName`, `signUp()`).
 - o **Classes:** Use PascalCase for class names (e.g., `User`, `ContentManager`).
 - o **Constants:** Use all uppercase with underscores for constants (e.g., `MAX_CONTENT_SIZE`, `DEFAULT_TIMEOUT`).
2. **Formatting and Indentation:**
 - o Use consistent indentation (typically 4 spaces or tabs).
 - o Place braces on a new line for functions and control structures (if, for, while).
 - o Maintain a clear and readable structure, avoiding overly long lines (typically limit lines to 80-120 characters).
3. **Comments and Documentation:**
 - o Use comments to explain complex logic, algorithms, or non-obvious code.
 - o Document functions, classes, and major sections of code using docstrings (following conventions such as Python's PEP-257).
 - o Update comments and documentation when code changes to ensure accuracy.
4. **Code Organization:**
 - o Organize code into logical modules and packages based on functionality.
 - o Separate concerns by following principles like MVC (Model-View-Controller) or similar patterns relevant to the project architecture.
5. **Error Handling:**
 - o Implement consistent error handling practices throughout the codebase.
 - o Use appropriate exception types and provide informative error messages for debugging and troubleshooting.
6. **Testing and Quality Assurance:**
 - o Write unit tests for critical functions and modules to ensure functionality and reliability.
 - o Conduct code reviews to enforce coding standards, identify potential issues, and share knowledge among team members.
7. **Version Control:**
 - o Follow best practices for version control (e.g., using Git) such as meaningful commit messages, branching strategies, and merging code changes.
8. **Performance Considerations:**
 - o Optimize code for performance where necessary, avoiding premature optimization.
 - o Profile code to identify and address bottlenecks and inefficiencies.
9. **Security:**
 - o Adhere to secure coding practices to mitigate risks such as SQL injection, cross-site scripting (XSS), and sensitive data exposure.
 - o Validate input data, sanitize outputs, and use parameterized queries in database interactions.
10. **Frameworks and Libraries:**
 - o Follow guidelines and conventions specific to frameworks and libraries used in the project (e.g., Django for Python web development).

By adhering to these coding standards, the GAppMarket project ensures that the codebase is consistent, maintainable, and scalable. It fosters collaboration among developers, enhances code readability, and facilitates efficient debugging and troubleshooting. Regular review and enforcement of these standards contribute to the overall quality and longevity of the project.

4.2 Screenshots (I developed this Website is Developed with Mobile First approach)







The screenshot shows the 'Contact Us' screen of the GAppMarket app. At the top, the title 'GAppMarket' is displayed above a three-line menu icon. Below the title, the word 'Contact Us' is centered in large, bold, black font. The screen contains four input fields with placeholder text: 'Name' (placeholder: 'Enter your name'), 'Email' (placeholder: 'Enter your email'), 'Subject' (placeholder: 'Enter the subject'), and 'Message' (placeholder: 'Enter your message'). At the bottom of the screen is a blue rounded rectangular button labeled 'Send Message'.

GAppMarket

Contact Us

Name

Ghansyam

Email

vajaghanshyam11@gmail.com

Subject

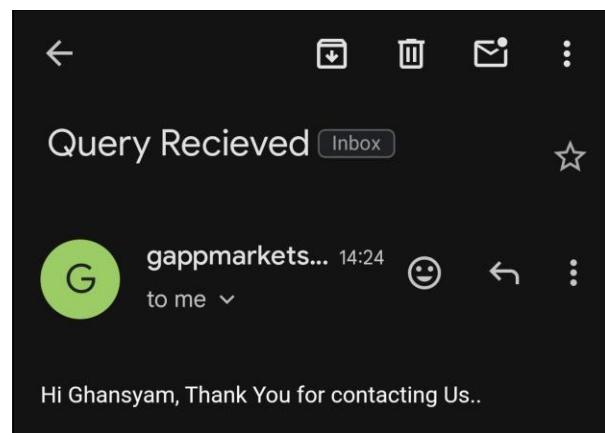
Testing

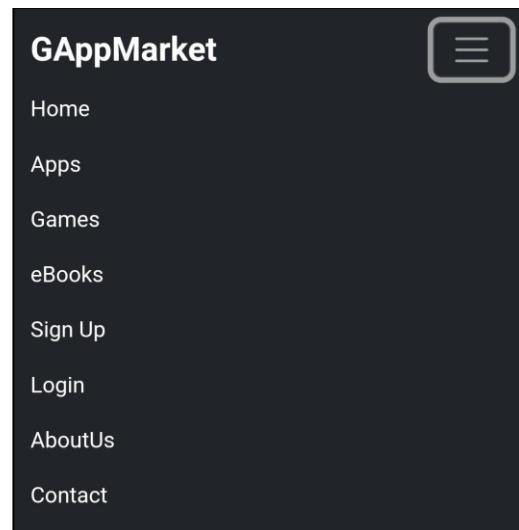
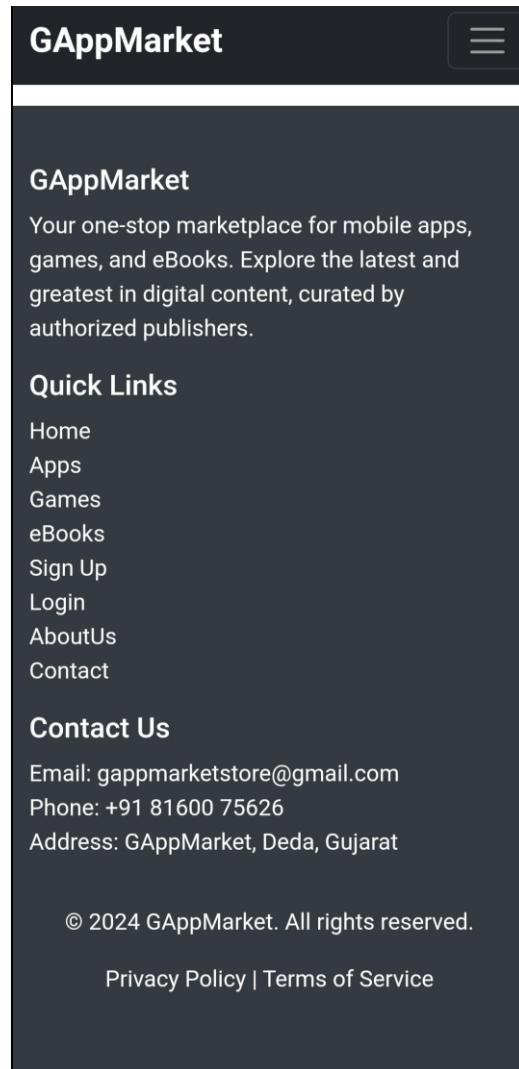
Message

Testing Continue|

Send Message

The screenshot shows a contact form titled "Contact Us" from "GAppMarket". The form includes fields for Name, Email, Subject, and a large Message area. The Name field contains "Ghansyam", the Email field contains "vajaghanshyam11@gmail.com", and the Subject field contains "Testing". The Message field contains "Testing Continue|". A blue rounded rectangle highlights the Message input field. At the bottom is a blue "Send Message" button.





The screenshot shows the 'About Us' section of the GAppMarket mobile application. At the top, the title 'GAppMarket' is displayed next to a menu icon. Below the title, the heading 'About Us' is centered. A detailed paragraph explains the project's purpose: creating a one-stop marketplace for mobile apps, games, and eBooks curated by authorized publishers. Underneath this, there is a section titled 'Developer Information' containing the developer's name, Ghanshyam Vaja, and the project's purpose, which is the MCA SEM-3 Software Project-2 Submission.

GAppMarket

Welcome to GAppMarket! This project is created by Ghanshyam Vaja as part of the MCA SEM-3 Software Project-2 Subject Submission. Our goal is to provide a one-stop marketplace for mobile apps, games, and eBooks, curated by authorized publishers.

Developer Information

Name: Ghanshyam Vaja

Project Purpose: MCA SEM-3 Software Project-2 Submission

The screenshot shows the 'Privacy Policy' section of the GAppMarket mobile application. At the top, the title 'GAppMarket' is displayed next to a menu icon. Below the title, the heading 'Privacy Policy' is centered. A paragraph explains the company's commitment to protecting user privacy and how they collect, use, and share information. The 'Information We Collect' section details the types of information collected, including personal, usage, and device information. The 'Device Information' section is partially visible at the bottom.

GAppMarket

Your one-stop marketplace for mobile apps, games, and eBooks. Explore the latest and

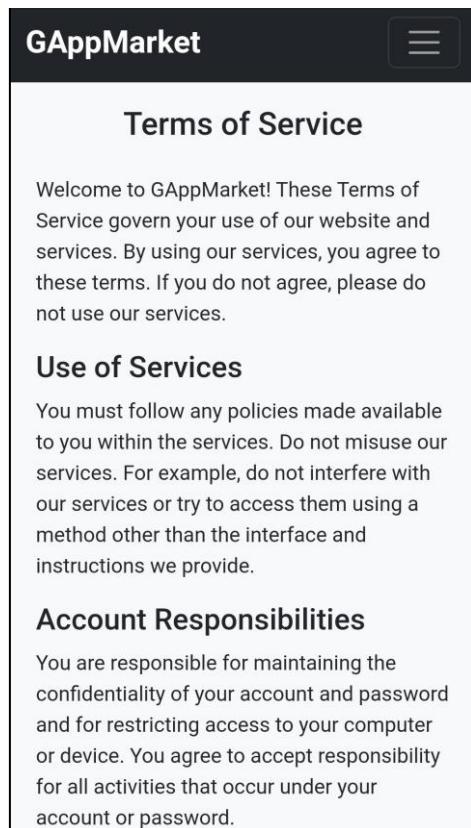
Privacy Policy

At GAppMarket, we are committed to protecting your privacy. This Privacy Policy explains how we collect, use, and share information about you when you use our website and services.

Information We Collect

We collect information about you when you provide it to us, when you use our services, and when other sources provide it to us. This includes:

- **Personal Information:** Information such as your name, email address, and other contact details.
- **Usage Information:** Information about your interactions with our services, such as the apps, games, and eBooks you download or access.
- **Device Information:** Information about the devices you use to access our



The screenshot shows the 'Terms of Service' page of GAppMarket. At the top, the 'GAppMarket' logo is displayed. Below it, the title 'Terms of Service' is centered. The main content area contains a welcome message and several sections: 'Use of Services' and 'Account Responsibilities'. Each section includes a detailed description of the terms.

GAppMarket

Terms of Service

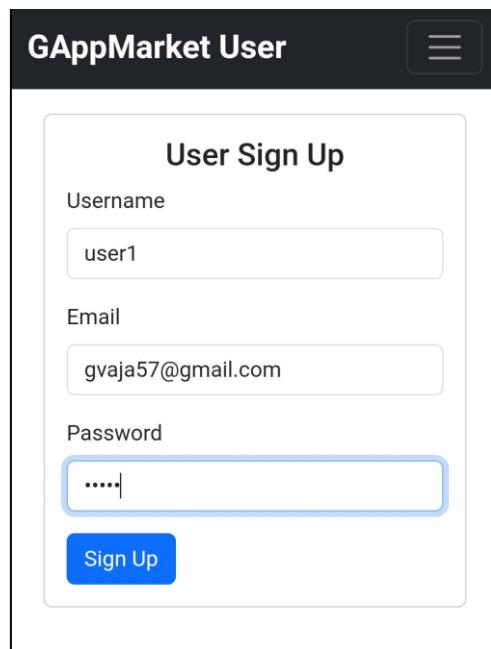
Welcome to GAppMarket! These Terms of Service govern your use of our website and services. By using our services, you agree to these terms. If you do not agree, please do not use our services.

Use of Services

You must follow any policies made available to you within the services. Do not misuse our services. For example, do not interfere with our services or try to access them using a method other than the interface and instructions we provide.

Account Responsibilities

You are responsible for maintaining the confidentiality of your account and password and for restricting access to your computer or device. You agree to accept responsibility for all activities that occur under your account or password.



The screenshot shows the 'User Sign Up' page of GAppMarket. The page has a dark header with the 'GAppMarket User' logo. The main form is titled 'User Sign Up' and contains three fields: 'Username', 'Email', and 'Password'. Each field has an input box. Below the form is a blue 'Sign Up' button.

GAppMarket User

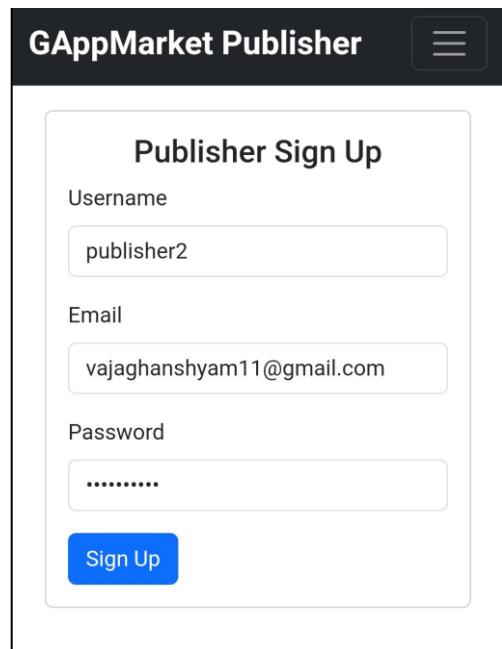
User Sign Up

Username

Email

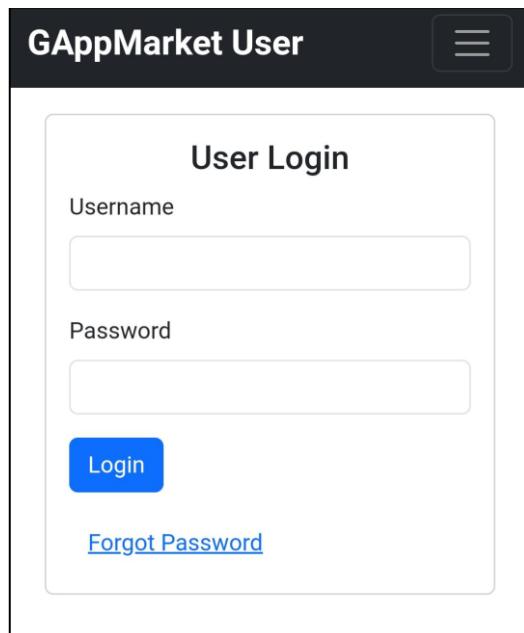
Password

Sign Up

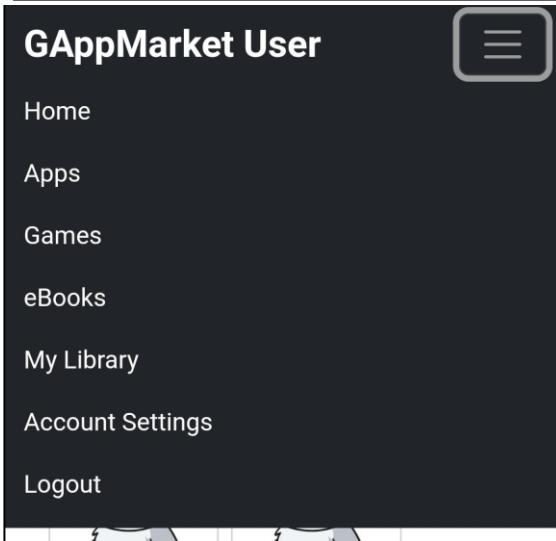
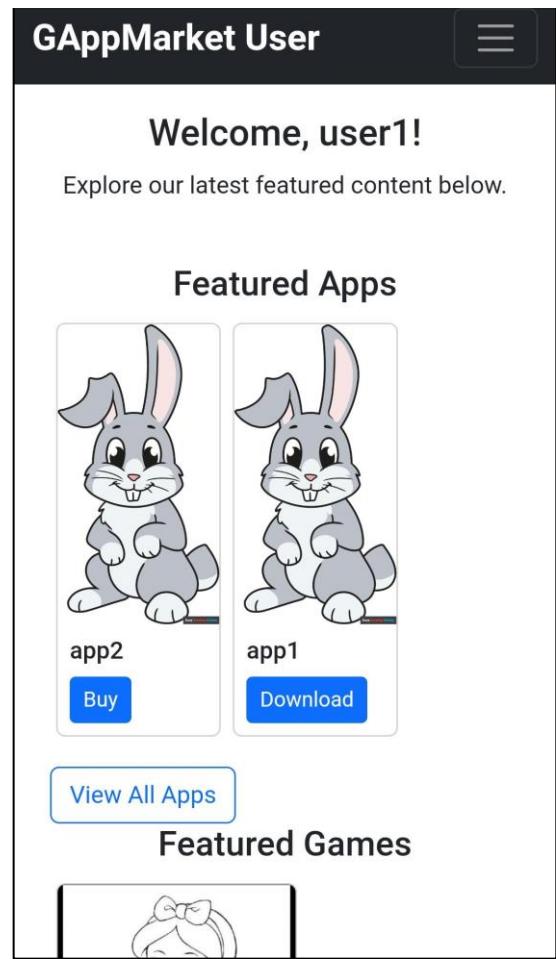


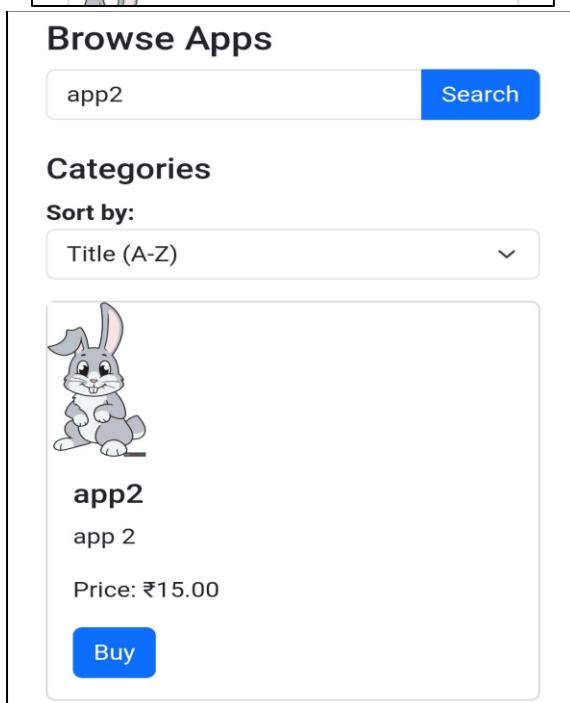
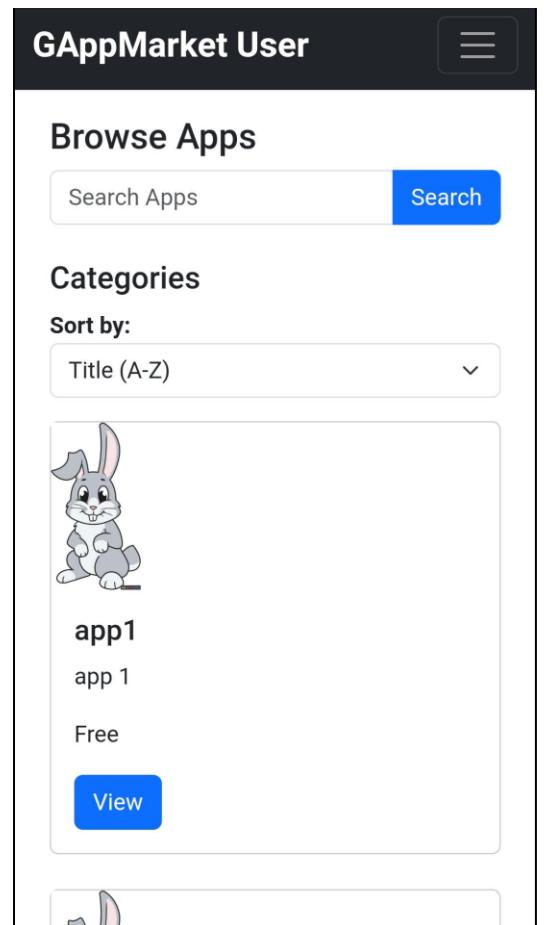
The screenshot shows the 'Publisher Sign Up' screen of the GAppMarket Publisher app. At the top, the title 'GAppMarket Publisher' is displayed next to a three-line menu icon. Below the title, the section 'Publisher Sign Up' is centered. The form contains three input fields: 'Username' with the value 'publisher2', 'Email' with the value 'vajaghanshyam11@gmail.com', and 'Password' represented by a series of dots. A blue 'Sign Up' button is located at the bottom right of the form.

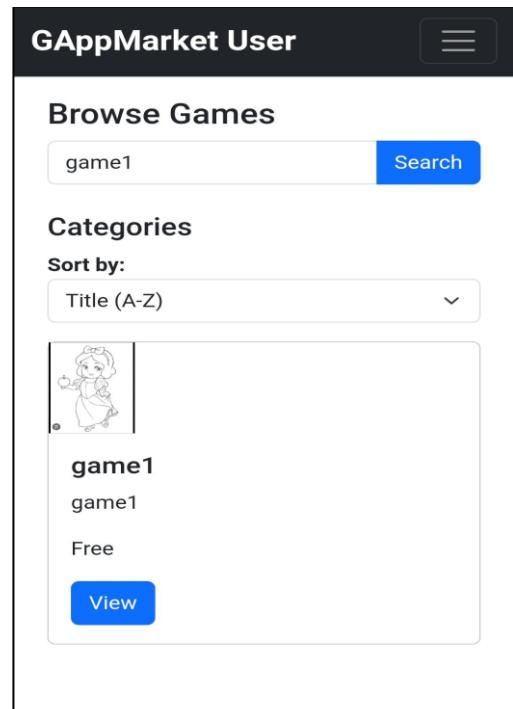
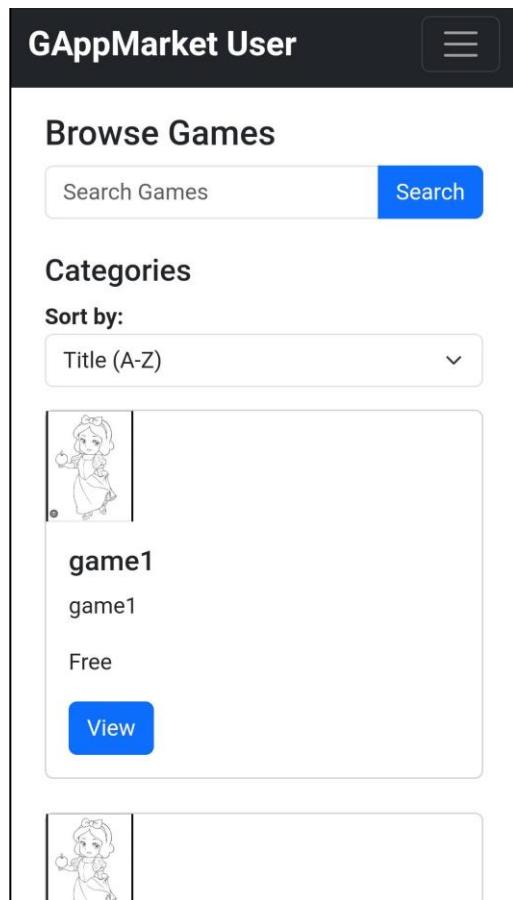
User Corner

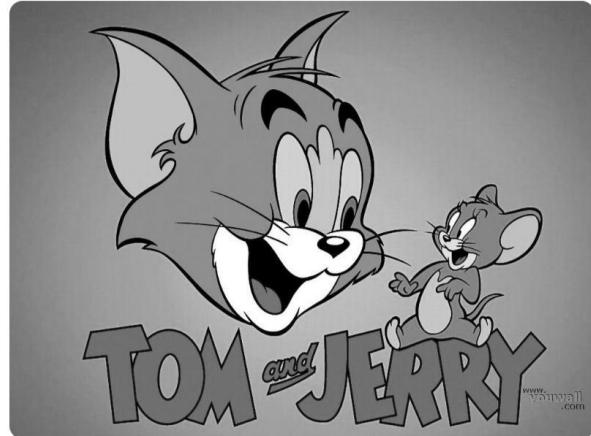


The screenshot shows the 'User Login' screen of the GAppMarket User app. At the top, the title 'GAppMarket User' is displayed next to a three-line menu icon. Below the title, the section 'User Login' is centered. The form contains two input fields: 'Username' and 'Password'. Below the password field is a blue 'Login' button. At the bottom of the form, there is a link labeled 'Forgot Password'.









Name : game1

Description : game1



[Download](#)

[Write a Review](#)

GAppMarket User



Name : ebook1

Description : ebook1



[Download](#)

[Write a Review](#)

Reviews

gvaja57 - 4 stars

Hello

Reviewed on July 7, 2024, 4:06 p.m.

GAppMarket User

≡

My Library

game2

game2

View Details

ebook1

ebook1

View Details

GAppMarket User

≡

Account Settings

Username

Email address

New Email address

New Password

Save Changes

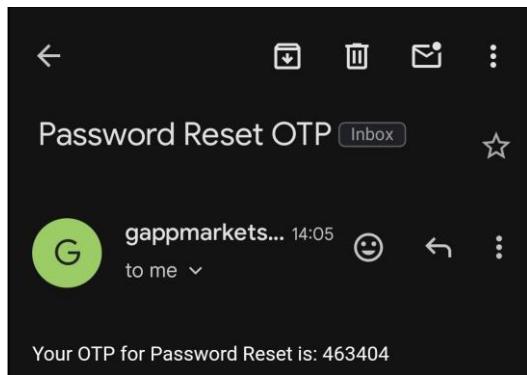
Delete Account

GAppMarket User

Forgot Password

Email

Send OTP



GAppMarket User

Verify OTP

OTP

Verify OTP

GAppMarket User

Change Password

New Password

Confirm Password

Change Password

Publisher Corner

GAppMarket Publisher

Publisher Login

Username
publisher1

Password
.....

Login

[Forgot Password](#)

GAppMarket Publisher

Total Content

6

Total Sales

2

Total Reviews

2

Recent Content

1. What is servlet? Explain life cycle of servlet (With suitable example.)	7 Marks
2. Write a servlet that generates plain text (Write and explain code)	7 Marks
3. Write a servlet that generates html response. (Write and explain code)	7 Marks
4. Differentiate HttpServlet and GenericServlet.	4 Marks
5. What is form data? Explain.	2 Marks
6. Explain the methods : getParameter(), getParameterValues()	7 Marks
7. List and explain any 10 HTTP request headers along with its methods.	7 Marks
8. How can we read form data in a servlet? Explain with example.	7 Marks
9. Explain any 7 standard CGI variables.	7 Marks
10. What is referer header? How it can be useful? Explain with suitable example.	5 Marks
11. List and explain any 10 response headers.	7 Marks
12. What are status codes? List and explain different status codes available.	7 Marks
13. How can we generate JPEG image from a servlet.	7 Marks
14. Explain the methods : sendRedirect() and sendRedirect()	7 Marks
15. What is the usage of cookies? Explain how cookies are sent and received.	7 Marks



A screenshot of the 'My Content' section. The title 'My Content' is at the top. Below it is a card for an app named 'app1' with the subtitle 'app 1'. The card contains a thumbnail image of Cinderella, a blue 'Edit' button, and a red 'Delete' button. Below this card is another card showing a thumbnail of Snow White holding an apple.

GAppMarket Publisher

Upload New Content

Title

Description

Logo
 Choose File No file chosen

Cover Image 1
 Choose File No file chosen

Cover Image 2
 Choose File No file chosen

Cover Image 3
 Choose File No file chosen

Cover Image 4
 Choose File No file chosen

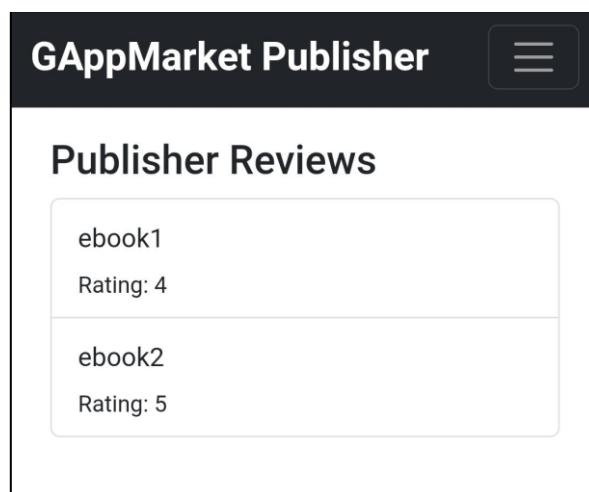
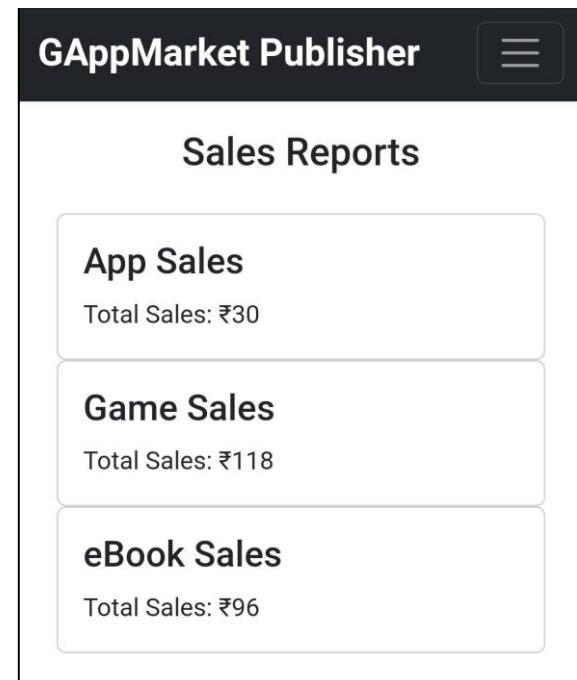
Content Type
 App

Type
 Free

Application File
 Choose File No file chosen

APK Version
 Enter APK Version

Upload Content



GAppMarket Publisher

Account Settings

Username
publisher1

Email address
ghanshyamvaja11@gmail.com

New Email address
vajaghanshyam12345@gmail.com

New Password

Save Changes

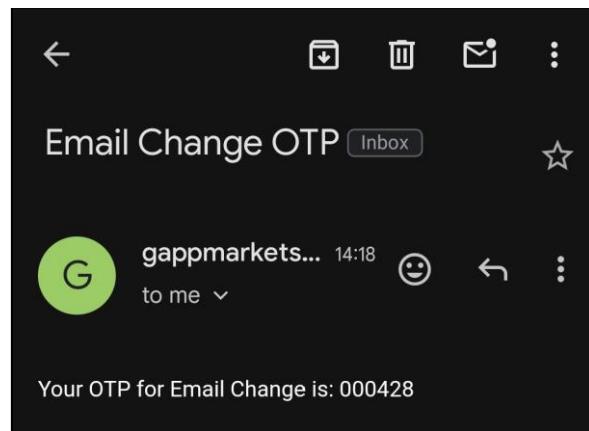
Delete Account

GAppMarket Publisher

Verify OTP

Enter OTP

Verify OTP



GAppMarket Publisher

Verify OTP

Enter OTP

000428

Verify OTP

GAppMarket Publisher

Account Settings

Username

publisher1

Email address

vajaghanshyam12345@gmail.com

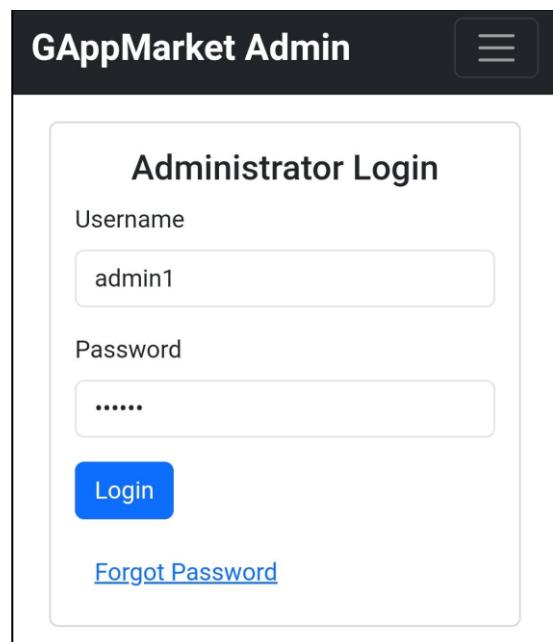
New Email address

New Password

Save Changes

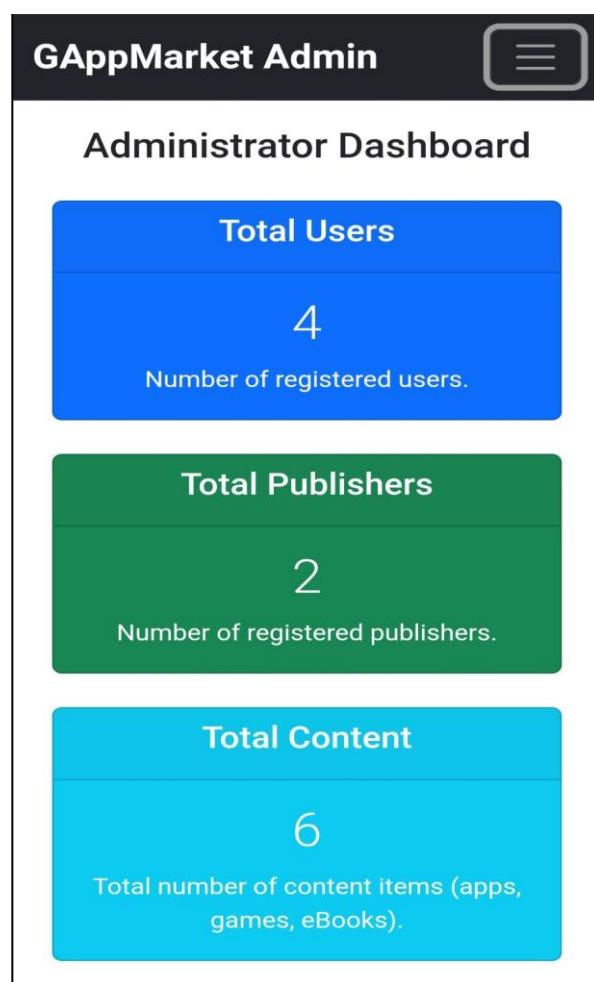
Delete Account

Administrator Corner



The screenshot shows the 'Administrator Login' screen of the GAppMarket Admin application. At the top, it displays the title 'GAppMarket Admin' and a three-line menu icon. Below this is a light gray rectangular form with rounded corners. It contains the following fields:

- Administrator Login** (Section header)
- Username**: A text input field containing the value 'admin1'.
- Password**: A text input field containing the value '.....'.
- Login**: A blue rectangular button with white text.
- [Forgot Password](#): A blue underlined link at the bottom right.

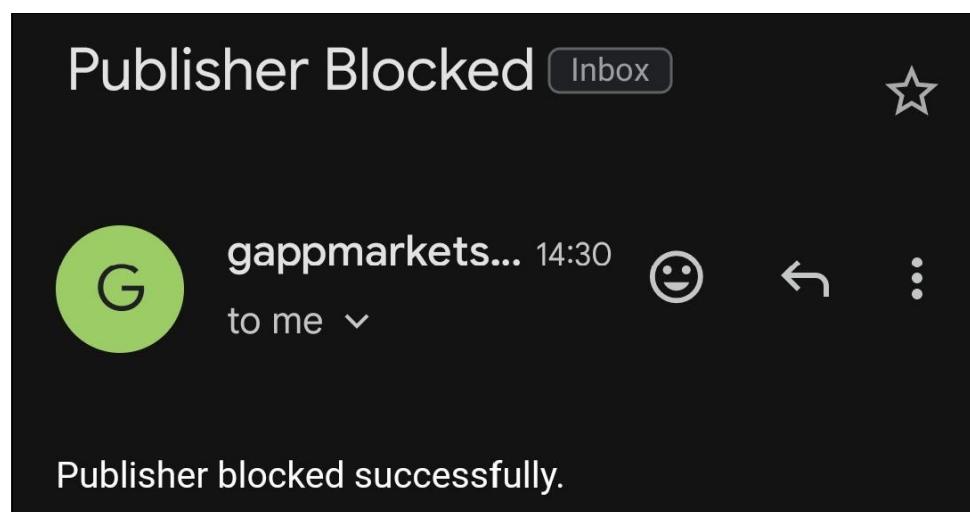


User Management					
ID	Username	Email	Status	Actions	
1	gvaja57	ghanshyamvaja11@gmail.com	Active	<button>Block</button>	<button>Delete</button>
2	user	vaja@gmail.com	Blocked	<button>Unblock</button>	<button>Delete</button>
3	user1	gvaja57@gmail.com	Active	<button>Block</button>	<button>Delete</button>
4	user2	vajaghanshyam11@gmail.com	Active	<button>Block</button>	<button>Delete</button>

User Management					
ID	Username	Email	Status	Actions	
1	gvaja57	ghanshyamvaja11@gmail.com	Blocked	<button>Unblock</button>	<button>Delete</button>
3	user1	gvaja57@gmail.com	Active	<button>Block</button>	<button>Delete</button>
4	user2	vajaghanshyam11@gmail.com	Active	<button>Block</button>	<button>Delete</button>

Publisher Management					
ID	Username	Email	Status	Actions	
5	publisher1	vajaghanshyam12345@gmail.com	Active	<button>Block</button>	<button>Delete</button>
6	publisher2	vajaghanshyam11@gmail.com	Active	<button>Block</button>	<button>Delete</button>

Publisher Management					
<input type="text" value="Search"/> <input type="button" value="Search"/>					
ID	Username	Email	Status	Actions	
5	publisher1	vajaghanshyam12345@gmail.com	Active	<input type="button" value="Block"/>	<input type="button" value="Delete"/>
6	publisher2	vajaghanshyam11@gmail.com	Blocked	<input type="button" value="Unblock"/>	<input type="button" value="Delete"/>



Content Management					
<input type="text" value="Search"/> <input type="button" value="Search"/>					
ID	Title	Publisher	Status	Actions	
14	app1	publisher1	Active	<input type="button" value="Block"/>	<input type="button" value="Delete"/>
15	app2	publisher1	Active	<input type="button" value="Block"/>	<input type="button" value="Delete"/>
16	game1	publisher1	Active	<input type="button" value="Block"/>	<input type="button" value="Delete"/>
17	game2	publisher1	Active	<input type="button" value="Block"/>	<input type="button" value="Delete"/>
20	ebook1	publisher1	Active	<input type="button" value="Block"/>	<input type="button" value="Delete"/>
21	ebook2	publisher1	Active	<input type="button" value="Block"/>	<input type="button" value="Delete"/>

GAppMarket Admin

Content Management

Search

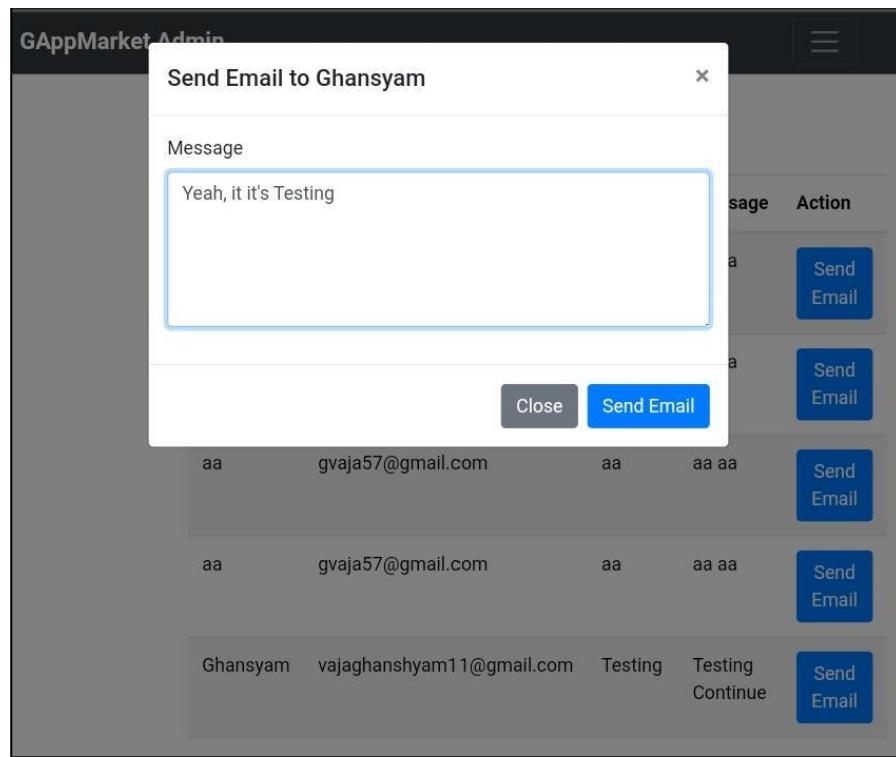
Search

ID	Title	Publisher	Status	Actions	
14	app1	publisher1	Blocked	<button>Unblock</button>	<button>Delete</button>
15	app2	publisher1	Active	<button>Block</button>	<button>Delete</button>
16	game1	publisher1	Active	<button>Block</button>	<button>Delete</button>
17	game2	publisher1	Active	<button>Block</button>	<button>Delete</button>
20	ebook1	publisher1	Active	<button>Block</button>	<button>Delete</button>
21	ebook2	publisher1	Active	<button>Block</button>	<button>Delete</button>

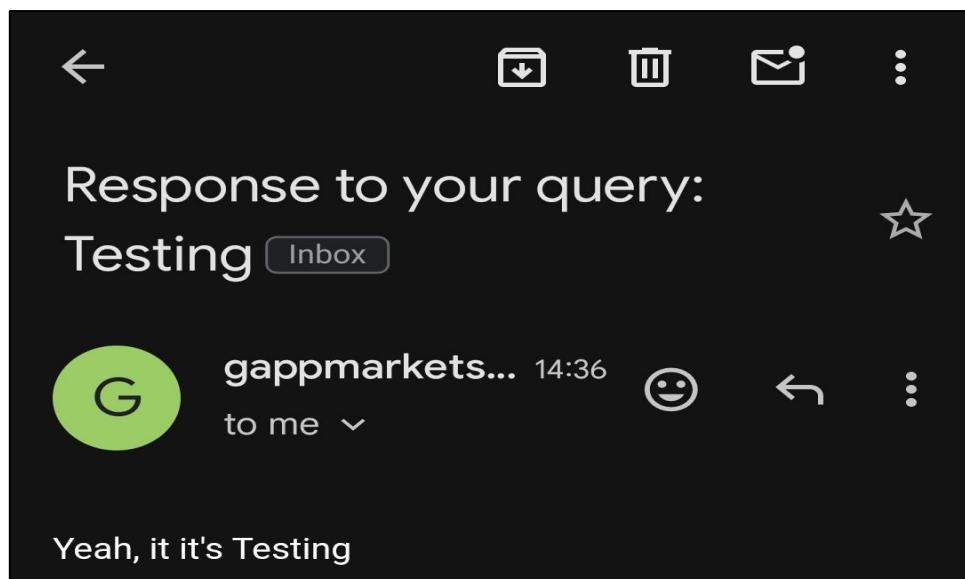
GAppMarket Admin

Contact Management

Name	Email	Subject	Message	Action
aa	gvaja57@gmail.com	ss	aa aa	<button>Send Email</button>
aa	gvaja57@gmail.com	aa	aa aa	<button>Send Email</button>
aa	gvaja57@gmail.com	aa	aa aa	<button>Send Email</button>
aa	gvaja57@gmail.com	aa	aa aa	<button>Send Email</button>
Ghansyam	vajaghanshyam11@gmail.com	Testing	Testing Continue	<button>Send Email</button>



Contact Management				
Name	Email	Subject	Message	Action
aa	gvaja57@gmail.com	ss	aa aa	<button>Send Email</button>
aa	gvaja57@gmail.com	aa	aa aa	<button>Send Email</button>
aa	gvaja57@gmail.com	aa	aa aa	<button>Send Email</button>
aa	gvaja57@gmail.com	aa	aa aa	<button>Send Email</button>



The screenshot shows the "GAppMarket Admin" interface. The top navigation bar has a menu icon. The main section is titled "Send Announcements". It includes fields for "Subject" (set to "Testing3") and "Message" (containing "Abcde"). There are checkboxes for "Send to Publishers" (unchecked) and "Send to Users" (checked). A blue "Send Announcement" button is visible. Below this, the "Recent Announcements" section lists "Testing2" (sent on July 8, 2024, 2:43 p.m.) and "4" (sent on July 7, 2024, 1:24 a.m.).

GAppMarket Admin

Account Settings

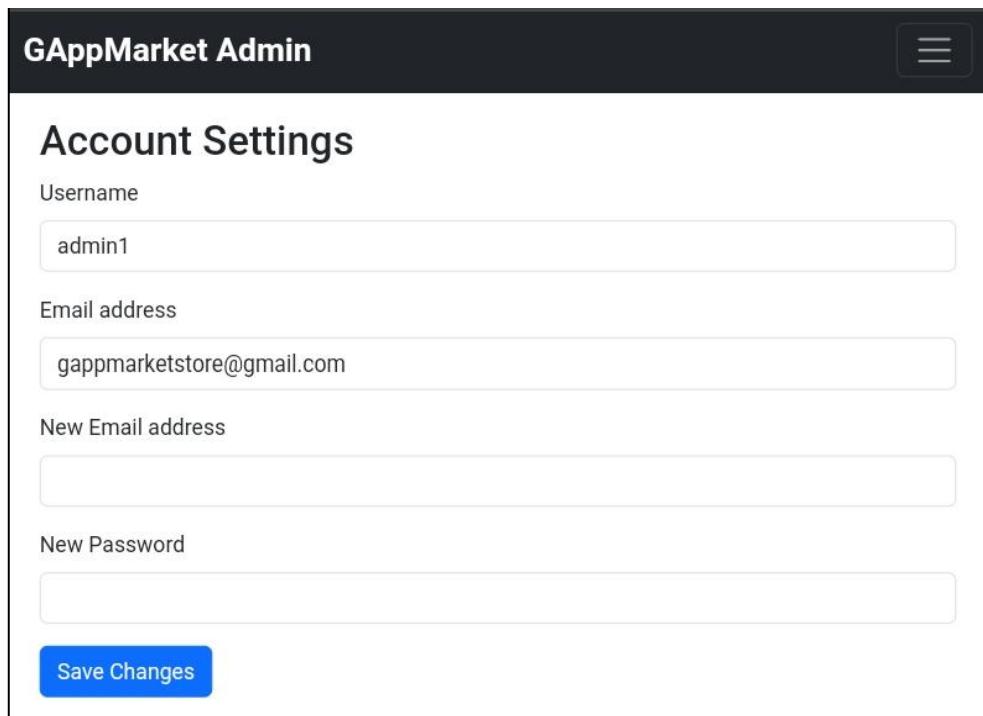
Username

Email address

New Email address

New Password

Save Changes



5. Agile Documentation

Agile methodologies emphasize iterative development, flexibility, and collaboration. The following Agile documentation components are essential for managing and guiding the GAppMarket project:

5.1 Agile Project Charter

The Agile Project Charter outlines the project's vision, objectives, scope, and initial expectations. It typically includes:

- **Project Vision:** A brief statement describing the overall purpose and goals of the GAppMarket project.
- **Objectives:** Specific, measurable outcomes the project aims to achieve (e.g., launch a scalable digital content marketplace).
- **Scope:** Defined boundaries and deliverables of the project, including initial features and functionalities.
- **Key Stakeholders:** Identification of stakeholders and their roles in the project.
- **Constraints and Assumptions:** Factors that may impact project delivery, such as budget constraints or technological dependencies.

Agile Project Charter - GAppMarket	
Vision	Scalable, user-friendly digital content marketplace for apps, games, and eBooks.
Objectives	<ul style="list-style-type: none">• Launch robust marketplace• User-friendly interface• Publisher content management• Secure payments
Scope	<ul style="list-style-type: none">• User registration/login• Content browsing/search• Purchase/download• Publisher dashboard• Admin panel
Stakeholders	<ul style="list-style-type: none">• Owner: Ghanshyam Vaja• Users: Consumers, Publishers
Constraints	<ul style="list-style-type: none">• Budget limits features• Third-party dependencies

5.2 Agile Roadmap / Schedule

The Agile Roadmap or Schedule outlines the high-level timeline and milestones for the project. It includes:

- **Phases or Iterations:** Breakdown of project development into phases or iterations (e.g., planning, development, testing).
- **Key Milestones:** Major checkpoints or deliverables throughout the project timeline (e.g., prototype completion, beta testing).
- **Dependencies:** Identification of critical dependencies that may affect scheduling or resource allocation.
- **Resource Allocation:** Allocation of team members, tools, and other resources required for each phase or milestone.

Agile Roadmap - GAppMarket

Phases or Iterations

- Planning
- Development
- Testing
- Launch

Key Milestones

Milestone	Target Date
Project Kickoff	June 26, 2024
Prototype Completion	July 6, 2024
Beta Testing	July 7, 2024
Official Launch	July 7, 2024

Dependencies

- Hosting service setup
- Content provider agreements

Resource Allocation

Phase	Team Member	Tools/Resources
Planning	Me-->Project Manager, Lead Developer, Business Analyst	Project Management Software
Development	Me-->Developer, UI/UX Designer	Development Tools, Design Software
Testing	Me	Testing Tools
Launch	Me	Hosting Services

5.3 Agile Project Plan

The Agile Project Plan details specific activities, tasks, and responsibilities for executing the project. It includes:

- **Task Breakdown:** Detailed breakdown of tasks required to achieve project objectives.
- **Task Assignments:** Assignment of tasks to team members, including estimated timeframes and dependencies.
- **Risk Management:** Identification and mitigation strategies for potential risks and issues.
- **Communication Plan:** Plan for regular communication, meetings, and updates within the Agile team.

Agile Project Plan - GAppMarket

Task Breakdown & Assignments

Task	Assigned To	Completed Date
Project Planning	Ghanshyam Vaja	June 26, 2024
HTML Structure Design	Ghanshyam Vaja	June 30, 2024
Django Models Creation	Ghanshyam Vaja	July 2, 2024
Publisher Dashboard UI	Ghanshyam Vaja	July 3, 2024
Administrator Views	Ghanshyam Vaja	July 4, 2024
Announcement Timezone Setup	Ghanshyam Vaja	July 6, 2024

Risk Management

- Budget constraints
- Third-party dependencies
- Technical challenges

5.4 Agile User Story (Minimum 3 Tasks)

Agile User Stories describe desired features or functionalities from an end-user perspective. Each User Story typically includes:

- **Title:** Brief description of the user story (e.g., "As a User, I want to search and purchase apps").
- **Description:** Detailed narrative from the user's viewpoint, outlining specific requirements or tasks.
- **Acceptance Criteria:** Criteria that must be met for the user story to be considered complete and ready for release.
- **Example User Stories:**
 1. As a User, I want to register and create an account to access the GAppMarket platform.
 2. As a Publisher, I want to upload apps, games, and eBooks with detailed descriptions and images.

- As an Administrator, I want to manage user accounts and monitor content quality and compliance.

Agile User Stories - GAppMarket

Title: User Registration

Description: As a User, I want to register and create an account to access the GAppMarket platform.

Acceptance Criteria:

- User can register with valid information.
- System validates and confirms registration.

Title: Content Upload

Description: As a Publisher, I want to upload content with details and images.

Acceptance Criteria:

- Publisher can upload files and enter details.

Title: Account Management

Description: As an Administrator, I want to manage user accounts and monitor content quality.

Acceptance Criteria:

- Admin can view user accounts.
- Admin can monitor content for quality.

5.5 Agile Release Plan

The Agile Release Plan outlines the schedule and content of upcoming releases. It includes:

- **Release Dates:** Planned dates for each release, based on prioritized features and milestones.
- **Feature Prioritization:** Prioritization of features or user stories for each release based on business value and dependencies.
- **Resource Allocation:** Allocation of resources (team members, testing environments) required for release preparation.
- **Risk Assessment:** Identification of potential risks that may impact release schedules and contingency plans.

Agile Release Plan - GAppMarket

Release Dates

Planned dates for the release, reflecting work from June 26 to July 7, 2024.

- Release 1: June 26, 2024 - July 7, 2024

Feature Prioritization

Features completed during the release period.

- User Registration
- Content Upload
- Account Management
- Dashboard for Publisher
- Content Detail Page
- Integration of Publisher-Specific Menus

Resource Allocation

Resources utilized for development.

- Team Member: Ghanshyam Vaja
- Testing: Self-testing on Development Environment

Risk Assessment

Identified risks and mitigation strategies.

- Risk 1: Feature completion delays
- Risk 2: Integration issues with new features
- Contingency Plan: Regular progress checks and adjustments

5.6 Agile Sprint Backlog

The Agile Sprint Backlog details the tasks, user stories, and activities planned for a specific sprint. It includes:

- **Selected User Stories:** User stories selected for implementation during the sprint, based on prioritization.
- **Task Breakdown:** Breakdown of tasks required to complete each user story, including estimates and dependencies.
- **Daily Standup Topics:** Topics for daily stand-up meetings, focusing on progress, challenges, and adjustments.

Agile Sprint Backlog - GAppMarket

Selected User Stories

User stories selected for implementation during the sprint.

- User Registration
- Content Upload
- Account Management

Task Breakdown

Tasks breakdown for each user story, including estimates and dependencies.

- User Registration:
 - Create registration form
 - Implement backend validation
 - Design user interface
 - Set up user database schema
 - Implement email verification
- Content Upload:
 - Develop upload functionality
 - Handle file storage and retrieval
 - Integrate with user accounts
 - Implement file format validation
 - Optimize file upload performance
- Account Management:
 - Implement user login functionality
 - Allow user profile editing
 - Implement password reset functionality
 - Implement user role management

5.7 Agile Test Plan

The Agile Test Plan outlines the approach and strategies for testing throughout the project. It includes:

1. **Testing Objectives:** Goals and objectives of testing, such as functional testing, integration testing, and user acceptance testing.

2. **Test Strategy:** High-level approach to testing, including tools, environments, and methodologies (e.g., automated testing).
3. **Test Cases:** Specific test cases for validating user stories and functionalities, including expected outcomes and criteria for success.
4. **Defect Management:** Procedures for reporting, tracking, and resolving defects identified during testing.

Agile Test Plan - GAppMarket

Testing Objectives

- Functional, Integration, and User Acceptance Testing.

Test Strategy

- Automated and Manual Testing in staging and production environments.

Test Cases

- Validate user registration, content upload, and account management functionalities.

Defect Management

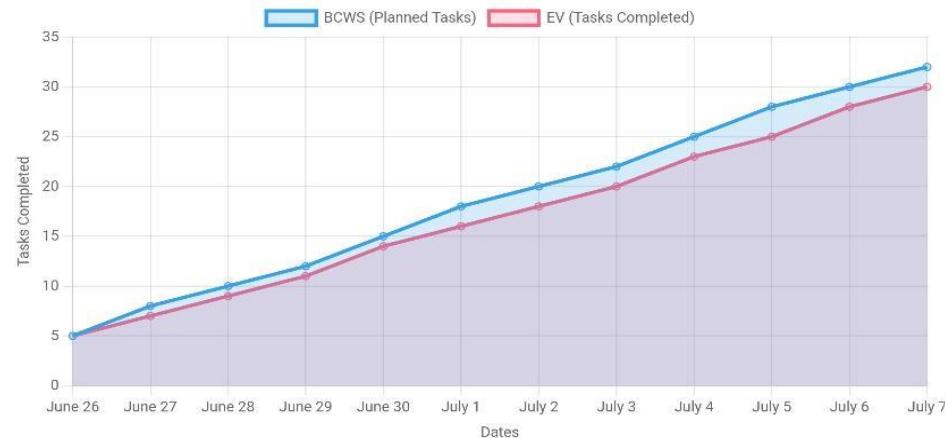
- Report, track, and resolve defects using bug tracking tools.

5.8 Earned-Value and Burn Charts and Gantt Chart

Earned-Value and Burn Charts are used to monitor and visualize project progress and performance:

- **Earned-Value Chart:** Compares planned work (budgeted cost of work scheduled - BCWS) against actual work completed (earned value - EV) and actual costs incurred (actual cost of work performed - ACWP). It provides insights into cost and schedule variances.

Earned-Value Chart - GAppMarket (June 26 to July 7)



- **Burn Charts:** Visualizes the progress of work completed (burn-down chart) or work remaining (burn-up chart) over time during sprints or the entire project. It helps track team velocity and predict project completion.

Burn-Down Chart - GAppMarket



- **Gantt Chart:** The Gantt chart above illustrates the duration and overlapping of tasks from June 26 to July 7 for the GAppMarket project. Each row represents a specific task with its start and end dates, showing the timeline of tasks and their durations.

Task Name	Jun	Jul
Task 1: Project Planning		
Task 2: UI Design		
Task 3: Backend Development		
Task 4: Testing and QA		

6. Proposed Enhancements

Proposed Enhancements highlight future improvements and features planned for the GAppMarket project to enhance functionality, user experience, and overall performance. Key areas of focus may include:

- **Enhanced User Interface:** Implementing a modern and responsive design to improve usability across devices.
- **Advanced Search and Filtering:** Enhancing search functionality with advanced filters based on categories, ratings, and user preferences.
- **Personalized Recommendations:** Introducing AI-driven algorithms to provide personalized app recommendations based on user behavior and preferences.
- **Enhanced Publisher Dashboard:** Adding analytics and insights to the publisher dashboard for better content management and performance monitoring.
- **Social Sharing and Integration:** Integrating social media sharing and login options to increase user engagement and platform visibility.
- **Enhanced Security Features:** Strengthening security measures to protect user data and transactions, including two-factor authentication and data encryption.

7. Conclusion

The Conclusion summarizes the key findings and outcomes of the GAppMarket project, highlighting its achievements and impact. It includes:

- **Project Accomplishments:** Recap of goals achieved and milestones reached during the project lifecycle.
- **User Feedback and Adoption:** Insights into user feedback, adoption rates, and satisfaction levels.
- **Challenges and Learnings:** Lessons learned, challenges overcome, and improvements made throughout the project.
- **Future Outlook:** Opportunities for future growth, enhancements, and potential expansions of the GAppMarket platform.

8. Bibliography

The Bibliography lists all the sources referenced or consulted during the development and documentation of the GAppMarket project. It includes:

1. Official Documentation
2. OpenAI