ASTANA IT UNIVERSITY

SE-2325

Welcome To

Game Log - Online Gaming Marketplace

Discover, Play, and Dominate in the world of gaming.

Team Members:
Oraz Khairulla
Seitkhan Nurmukhammed
Shyntasov Abdurahman

Git Hub:
https://github.com/OrazKhairulla/Online-Gaming-Marketplace.git

Project Goals and Objectives

Goal:

Develop a platform for buying, selling, and trading computer games, focusing on the secondary market and integration with user libraries.

Project Goals and Objectives

Objectives:

- Design and implement a database (MongoDB) to store information about games, users, orders, and shopping carts.
- Develop a REST API using Go (Gin Gonic) for frontend and backend communication.
- Create a user interface (HTML, CSS, JavaScript) with features for browsing, searching, and filtering games.
- Implement user registration, authentication (JWT), adding games to cart, placing orders, and viewing game libraries.
- Ensure basic application security (password hashing, protection against NoSQL injections).
- Conduct application testing.

GAME LOG

Home All Games **Cart**

Market Overview and Competitive Advantages

Existing platforms (Steam, GOG, Epic Games Store) have limitations:

- Steam: No secondary market for games.
- GOG: Smaller game selection compared to Steam.
- Classified ad sites: No specialization, lack of security guarantees, no library integration.

Our Project:

- Focuses on the secondary market.
- Provides a user-friendly, specialized interface.
- Integrates with user game libraries.

GAME LOG

Home All Games **Cart**

Application Architecture

Technology Choices:

Frontend: Maximum control over code, minimized dependencies.

Backend: Go + Gin - high performance and ease of development.

Deployment: Docker for containerization for easy deployment and scalability.

MongoDB: NoSQL, schema flexibility, scalability.

Frontend (HTML, CSS, JavaScript) ↔ API (Go, Gin Gonic) ↔ Backend (Go) ↔ Database (MongoDB)

GAME LOG

Home All Games **Cart**

User Interface

Technologies: HTML, CSS, JavaScript

- User registration and authentication.
- Game browsing with search and filtering.
- Adding games to the shopping cart.
- Viewing and editing the shopping cart.
- Placing orders.
- Viewing the library of purchased games.
- Editing user profile.

Responsive design for various devices.

Backend

Technologies: Go, Gin Gonic.

- Golang: High-performance backend development.
- Gin Framework: Lightweight web framework for Go.
- MongoDB: NoSQL database for efficient data storage and retrieval.

API (Main Routes):

- POST /api/auth/register: Registration
- POST /api/auth/login: Login
- GET /api/games/getall: Get all games
- GET /api/games/search: Search games
- POST /api/cart: Add to cart (protected)
- GET /api/cart: Get Cart
- DELETE /api/cart: Remove from cart (protected)
- POST /api/orders: Place Order
- GET /api/orders: Get Orders
- POST /api/user/update: update user info

Authentication: JWT (JSON Web Token).

Password Hashing: bcrypt.

Security Features

Keep Your Website Safe and Secure

- Password Hashing: bcrypt.
- Authentication: JWT (JSON Web Token).
- Protection against NoSQL injections: Using bson.M and bson.D for building MongoDB queries.
- Server-side data validation.

GAME LOG

Home All Games **Cart**

Results and Conclusions

Testing

Manual testing of core functionalities was performed and Unit testing of core API functions was performed (Go, testing package).

Our Service 01

- Registration and authentication. Adding/removing items from the cart.
- Placing an order.
- Viewing the library.
- Searching for games.
- Editing the profile

Our Service 02

- Registration and authentication functions.
- Shopping cart functions.
- Order placement functions.

Future development possibilities

Achieved Goals:

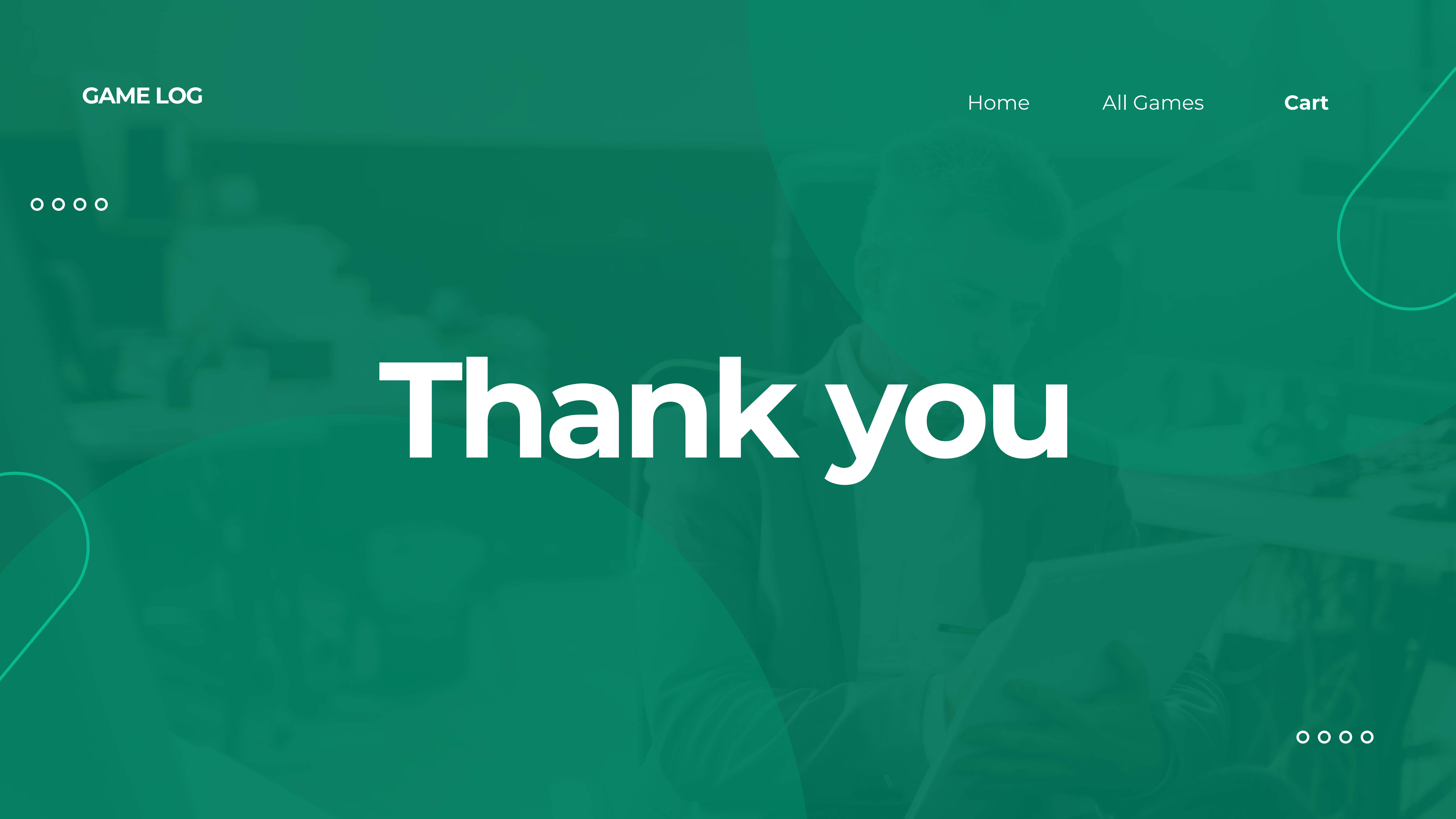
- Developed a platform for buying and selling games.
- Implemented core functionality (registration, login, cart, orders, library). Utilized modern technologies (Go, Gin, MongoDB, JWT).
- Implemented basic security measures.

Future Plans:

- Add pagination and filtering for games.
- Implement comprehensive testing (unit tests, load testing).
- Improve error handling.
- Advanced Filtering.

GAME LOG

Home All Games **Cart**



Thank you