ITE 18 – Application Development and Emerging Technologies

Web/Mobile Application – Final Project Submission

Student Name(s): A.J. Jandrei N. Mercado
Project Title: GameHaqqs2 **Course/Section:** ITE18-EJ1

Date of Submission: December 22-26, 2025

---

1. Project Overview

1.1 Project Description

GameHaqqs2 is a **full-stack web application** designed for gaming communities to manage games, track achievements, display leaderboards, and enable user interaction. The system consists of a **React-based frontend UI** and a **Laravel backend API**. The application solves the problem of fragmented game progress tracking by providing a centralized platform where users can view achievements, rankings, and game-related updates in one place.

1.2 Target Users

- Gamers who want to track achievements and rankings
- Community members who interact through comments and notifications
- Moderators who manage user-generated content
- Administrators who manage games, achievements, and users

1.3 Key Features

- User authentication and role-based access (Admin, Moderator, User)
- Game and achievement management
- Leaderboards and ranking system
- Community Posting, Commenting features
- React-based responsive user interface
- RESTful API integration between frontend and backend

1.4 Technology Stack

- **Frontend:** React.js, JavaScript, HTML5, CSS3
- **Backend:** Laravel (PHP)
- **Database:** MySQL
- **API:** RESTful JSON APIs
- **Tools:** Git, Composer, Node.js, npm, Laravel Artisan, Postman

2. App Plan

2.1 Project Scope

In Scope: - Full-stack web application (React frontend + Laravel backend)
- User authentication and authorization
- Game, achievement, leaderboard, and notification management - API-based communication

Out of Scope: - Native mobile applications
- Real-time multiplayer gameplay

2.2 Objectives & Goals

- Develop a user-friendly React frontend
- Build a secure and scalable backend API **2.3 User Stories & Use**

Cases :

User Story 1:
As a gamer, I want to view my achievements so that I can track my progress.

Acceptance Criteria:

- [✓] Achievements are displayed in the UI - [✓] Progress data is retrieved from the API

User Story 2:
As an admin, I want to manage games so that the platform remains updated.

Acceptance Criteria:

- [✓] Admin dashboard allows game creation and updates - [✓] Changes are reflected on the frontend

User Story 3:
As a moderator, I want to manage community posts to maintain friendly posts.

- [✓] Moderator dashboard approves and declines pending posts from users.

2.4 System Architecture

The system follows a **client–server architecture**. The React frontend handles user interaction, state management, and API requests. The Laravel backend processes requests, applies business logic, interacts with the database, and returns JSON responses. Communication occurs over HTTP using RESTful endpoints.

2.5 Development Timeline

Phase	Description	Timeline
Phase 1	Planning & UI Design	Week 1
Phase 2	Backend & Database Setup	Week 2
Phase 3	Frontend Development (React)	Week 3–4
Phase 4	Integration & Testing	Week 5

3. UI/UX Design

3.1 Design Philosophy

The UI follows a **modern, clean, and user-centered design** approach. Emphasis is placed on clarity, accessibility, and ease of navigation. Components are reusable and consistent across the application.

3.2 Color Scheme

- **Primary Color:** Dark Blue (#1E3A8A) – Used for headers and primary actions
- **Secondary Color:** Light Gray (#F3F4F6) – Used for backgrounds
- **Accent Color:** Green (#22C55E) – Used for success indicators
- **Text Color:** Dark Gray (#111827)

3.3 Typography

- **Font Family:** Sans-serif (e.g., Inter / Roboto)
- **Heading Size:** 24px–32px
- **Body Text Size:** 14px–16px
- **Line Height:** 1.5 **3.4 UI Components**
  - **Buttons:** Primary and secondary action buttons
  - **Input Fields:** Validated forms for login and data entry
  - **Navigation Bar:** Main navigation across pages
  - **Cards:** Used to display games, achievements, and leaderboards
  - **Modals:** Confirmation and alert dialogs **3.5 Wireframes & Mockups** GameHaqqs2

Wireframe:

https://www.figma.com/design/LlercIFdYRLpnvIx8Pmr40/GameHaqqs?node-id=01&t=NyARYBBB0EHN8WpP-1

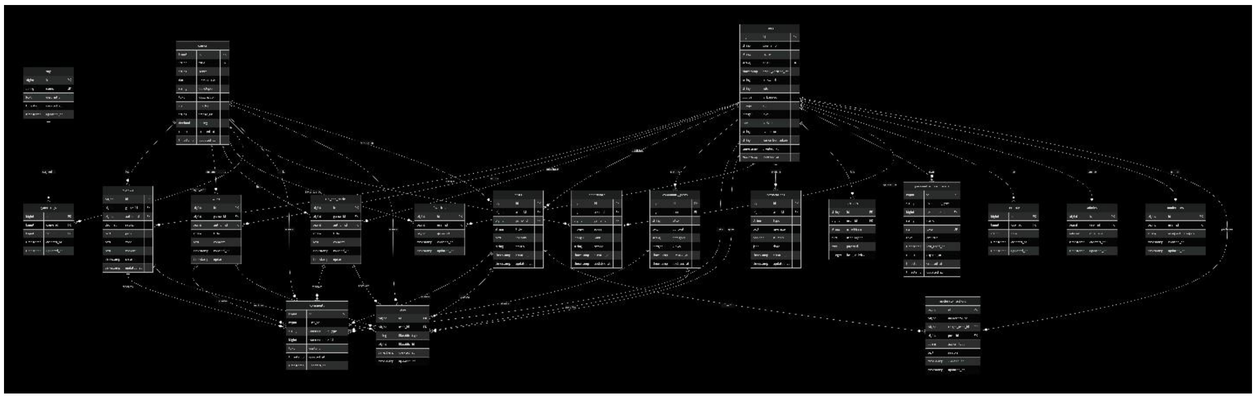
3.6 User Flows

- User Registration and Login Flow
- Viewing Games and Achievements Flow
- Leaderboard Viewing Flow

4. Database Architecture (ERD)

4.1 Entity Relationship Diagram

The ERD illustrates the database structure of the system, showing how core entities are interconnected to support the platform's functionality.



4.2 Entity Descriptions Users

- Stores registered user information, credentials, and roles
- Users can earn achievements, post comments, appear on leaderboards, and receive notifications **Roles**
- Defines user roles such as Admin, Moderator, and User
- Determines access and permissions within the system

Games

- Stores information about games available on the platform
- Serves as the basis for achievements and leaderboards

Achievements

- Represents milestones or goals linked to specific games
- Earned by users as part of gameplay progress

User Achievements

- Junction entity that links users and achievements
- Records achievement status and completion details **Leaderboards**
- Stores ranking data of users per game
- Used to display user performance and rankings

Comments

- Stores user-generated comments related to games or achievements
- Supports community interaction **Notifications**

- Stores system-generated messages sent to users
- Includes alerts such as achievement unlocks and updates

Moderation / Admin Actions

- Tracks actions performed by administrators or moderators
- Includes content management and user control activities

4.3 Relationships

- **Roles → Users** ○ One-to-Many (one role can be assigned to many users)
- **Users ↔ Achievements** ○ Many-to-Many (via User Achievements)
- **Games → Achievements** ○ One-to-Many (one game can have multiple achievements)
- **Users → Leaderboards** ○ One-to-Many (a user can have multiple leaderboard records)
- **Games → Leaderboards** ○ One-to-Many (a game can have multiple leaderboard entries)
- **Users → Comments** ○ One-to-Many (a user can create multiple comments)
- **Users → Notifications** ○ One-to-Many (a user can receive multiple notifications)
- **Users → Moderation / Admin Actions** ○ One-to-Many (admins or moderators can perform multiple actions)

4.4 Database Normalization

The database follows normalization up to **Third Normal Form (3NF)** to eliminate redundancy and ensure data integrity.

5. Application Features & Functionality

5.1 Feature 1: User Authentication & Registration

Description: Secure account system enabling users to register, log in, and maintain authenticated sessions.

Functionality:
• User registration and login with email and password
• Token-based authentication using Laravel Sanctum
• Secure password hashing and validation
• Session persistence with bearer tokens
• Logout with token invalidation
• Profile updates (name, email, password)
• Automatic redirect on token expiration

5.2 Feature 2: Games Library & Discovery

Description: Centralized game catalog for browsing and discovering games.

Functionality:
• Browse games with pagination
• View detailed game information
• Search and filter games
• Public game listings
• Game thumbnails and ratings display
• Sort games by popularity, rating, or date

5.3 Feature 3: Reviews & Ratings System

Description: Community-driven review and rating system for games.

Functionality:
• Create, edit, and delete reviews
• Rate games from 1–5 stars
• View and sort reviews
• Like and comment on reviews
• Display reviewer information
• Earn XP for reviews

5.4 Feature 4: Tips & Tricks Sharing

Description: Platform for sharing gameplay tips and strategies.

Functionality:
• Create, edit, and delete tips
• Browse tips per game
• Like and comment on tips
• View tip author details
• XP rewards for contributions

5.5 Feature 5: Favorites Management

Description: Personalized system for bookmarking games.

Functionality:
• Add or remove favorite games
• Toggle favorites from game pages
• View favorites in profile
• Persistent favorites across sessions

5.6 Feature 6: Community Posts & Discussions

Description: Social feature for sharing posts and engaging in discussions.

Functionality:
• Create and view community posts
• Like and comment on posts
• Delete own posts
• View post statistics
• Earn XP for participation

5.7 Feature 7: User Profile & Dashboard

Description: Dashboard displaying user stats, achievements, and activity.

Functionality:
• View XP, level, and achievements
• Track reviews, tips, and posts
• View favorite games
• Update profile information

• View leaderboard ranking

5.8 Feature 8: Leaderboard & Rankings

Description: Competitive ranking system based on user XP.

Functionality:
• View global leaderboard
• Rank users by XP
• View user levels and badges
• Compare rankings with others

5.9 Feature 9: Achievement System

Description: Reward system for milestones and engagement.

Functionality:
• View available and unlocked achievements
• Automatic achievement tracking
• Display badges on profiles
• Achievement notifications

5.10 Feature 10: Wiki & Game Guides

Description: Community-maintained game guides and documentation.

Functionality:
• Create, edit, and delete wiki articles
• Browse wikis by game
• Like and comment on wikis
• Earn XP for contributions

5.11 Feature 11: Comments & Engagement

Description: Interactive commenting system across all content.

Functionality:
• Comment on reviews, tips, posts, and wikis
• Edit and delete own comments
• Like comments

• Nested discussion threads

5.12 Feature 12: Notification Center

Description: Alert system for user activity and achievements.

Functionality:
• Notifications for likes, comments, and achievements
• Mark notifications as read
• View notification history
• Unread notification indicators

5.13 Feature 13: Search & Filter

Description: Advanced search for games, users, and content.

Functionality:
• Search games and users
• Filter and sort results
• Real-time search suggestions
• Paginated results

5.14 Feature 14: User Activity Tracking

Description: Activity log showing user contributions and engagement.

Functionality:
• View personal activity timeline
• Track reviews, tips, posts, and comments
• Display recent activity on profiles

5.15 Feature 15: Like & Interaction System

Description: Unified engagement system for community content.

Functionality:
• Like/unlike content across the platform
• Display like counts
• Highlight popular content
• Earn XP from engagement

6. Security & Error Handling

6.1 Security Measures Implemented

- Password hashing and secure authentication
- Role-based authorization
- Input validation on both frontend and backend

6.2 Error Handling

Errors are handled gracefully using frontend alerts and standardized backend JSON error responses.

7. Installation & Setup Instructions

7.1 Prerequisites

- Node.js & npm
- PHP 8+
- Composer
- MySQL

7.2 Installation Steps

1. Extract the project ZIP
2. Backend: Run `composer install` and configure `.env`
3. Frontend: Run `npm install` in the React directory
4. Run database migrations
5. Start backend and frontend servers

7.3 Running the Application

- Frontend: cd frontend npm run dev
  click: `http://localhost:5173`
- Backend:
  php artisan serve
  click: `http://127.0.0.1:8000`

8. Testing

8.1 Test Cases

- Login and authentication – Pass
- Game listing – Pass
- Leaderboard display – Pass
- Community Posting – Pass

- Writing Review, Tips and Tricks – Pass

8.2 Known Issues & Limitations

- No offline support
- Limited UI animations

9. Code Quality & Documentation

9.1 Code Structure

The project is separated into frontend (React) and backend (Laravel) directories following best practices.

Backend Structure:

```
GameHaqqs2/
|
 ├── app/
|   ├── Http/
|   |   ├── Controllers/
|   |   |  └── Api/
|   |   |       ├── AuthController.php         # User registration, login, logout
|   |   |       ├── UserController.php         # User profile, list, role management
|   |   |       ├── GameController.php          # Game CRUD, favorites
|   |   |       ├── ReviewController.php        # Review CRUD, likes, comments
|   |   |       ├── TipController.php          # Tips CRUD, likes, comments
|   |   |       ├── WikiController.php         # Wiki CRUD, likes, comments
|   |   |       ├── PostController.php         # Community posts CRUD
|   |   |       ├── CommentController.php       # Comment CRUD, likes
|   |   |       ├── TagController.php          # Tag management
|   |   |       ├── LeaderboardController.php    # Leaderboard rankings
|   |   |       ├── AchievementController.php    # Achievement system
|   |   |       ├── NotificationController.php   # User notifications
|   |   |       ├── AdminController.php         # Admin dashboard data
|   |   |       ├── ModeratorController.php      # Content moderation
|   |   |      └── ReportController.php        # User reports
|   |   |
|   |   └── Middleware/
|   |      └── RoleMiddleware.php            # Role-based access control
|   |
|   ├── Models/
```

```
│   │   ├── User.php                       # User model (Sanctum auth)
│   │   ├── Game.php                       # Game model
│   │   ├── Review.php                     # Review model
│   │   ├── TipAndTrick.php                # Tips model
│   │   ├── Wiki.php                       # Wiki model
│   │   ├── Post.php                       # Post model
│   │   ├── CommunityPost.php              # Community post model
│   │   ├── Comment.php                    # Comment model (polymorphic)
│   │   ├── Like.php                       # Like model (polymorphic)
│   │   ├── Favorite.php                   # Favorite model
│   │   ├── Tag.php                        # Tag model
│   │   ├── GameTag.php                    # Game-Tag pivot
│   │   ├── Achievement.php                # Achievement model
│   │   ├── UserAchievement.php            # User achievements
│   │   ├── Leaderboard.php                # Leaderboard model
│   │   ├── Notification.php               # Notification model
│   │   ├── Report.php                     # Report model
│   │   ├── Admin.php                      # Admin model
│   │   ├── Moderator.php                  # Moderator model
│   │   ├── ModeratorAction.php            # Moderator actions log
│   │   └── UserRole.php                   # User roles model │
│
│   ├── Policies/                          # Authorization policies
│   ├── Providers/                         # Service providers
│   └── Services/                          # Business logic services
│
├── config/
│   ├── app.php                            # Application config
│   ├── auth.php                           # Authentication config
│   ├── database.php                       # Database config
│   ├── sanctum.php                        # API auth config │
├── cors.php                               # CORS configuration │
└── ...
│
├── database/
│   ├── migrations/
│   │   ├── 2025_09_25_000001_create_users_table.php
│   │   ├── 2025_09_25_143529_create_games_table.php
```

```
|   |    ├── 2025_09_25_143530_create_reviews_table.php
|   |    ├── 2025_09_25_143531_create_tip_and_tricks_table.php
|   |    ├── 2025_09_25_143530_create_wikis_table.php
|   |    ├── 2025_10_08_000010_create_tags_and_pivots.php
|   |    ├── 2025_10_08_000020_create_favorites_likes_comments_notifications_roles.php
|   |    ├── 2025_10_09_000120_create_posts_table.php
|   |    ├── 2025_10_09_000130_create_leaderboard_table.php
|   |    ├── 2025_10_17_000001_create_comments_table.php
|   |    ├── 2025_10_17_000002_create_community_posts_table.php
|   |    ├── 2025_11_22_173247_create_reports_table.php |   |
 ├── 2025_11_24_084336_create_achievements_tables.php
|   |   └── ...
|   |
|   ├── seeders/
|   |   └── DatabaseSeeder.php                 # Sample data seeder
|   |
|   └── factories/                             # Model factories
|
 ├── routes/
|   ├── api.php                                # All API routes
|   ├── web.php                                  # Web routes (minimal)
|   └── console.php                              # Artisan commands |
 ├── storage/
|   ├── app/                                   # File storage
|   ├── logs/                                  # Application logs
|   └── framework/                               # Framework cache
|
 ├── .env                                      # Environment configuration
 ├── composer.json                                # PHP dependencies
└── artisan                                    # Laravel CLI tool
```

Frontend Structure:

```
frontend/

|

 ├── src/

|   ├── main.tsx                               # Application entry point
|   ├── App.tsx                        # Main app component, routing
|   ├── index.css                      # Global styles
|   |
|   ├── pages/                         # Page components
|   |   ├── LandingPage.tsx                # Homepage/landing
|   |   ├── RegisterPage.tsx               # User registration
|   |   ├── LoginPage.tsx                  # User login
|   |   ├── GamesLibrary.tsx               # Games browsing
|   |   ├── GameDetail.tsx                 # Single game view
|   |   ├── UserProfile.tsx                # User profile page
|   |   ├── UserDashboard.tsx              # User dashboard
|   |   ├── AdminDashboard.tsx             # Admin panel
|   |   ├── ModeratorDashboard.tsx         # Moderator panel
|   |   ├── TipsPage.tsx                   # Tips browsing
|   |   ├── WikiPage.tsx                   # Wiki pages
|   |   └── Settings.tsx                   # User settings
|   |
|   ├── components/                    # Reusable components
|   |   ├── Navbar.tsx                     # Navigation bar
|   |   ├── CommunityPosts.tsx             # Posts component
|   |   ├── NotificationDropdown.tsx       # Notifications UI
|   |   ├── PendingPostsModeration.tsx     # Moderation component
|   |   ├── ConfirmationDialog.tsx         # Confirm dialogs
|   |   ├── ErrorBoundary.tsx              # Error handling
|   |   |
|   |   ├── admin/                         # Admin components
|   |   |   ├── UserManagement.tsx
```

```
│   │   │   ├── GameManagement.tsx
│   │   │   └── Statistics.tsx
│   │   │
│   │   └── ui/                    # UI primitives (shadcn/ui)
│   │       ├── button.tsx
│   │       ├── card.tsx
│   │       ├── input.tsx
│   │       ├── dialog.tsx
│   │       ├── dropdown-menu.tsx
│   │       ├── table.tsx
│   │       ├── badge.tsx
│   │       ├── avatar.tsx
│   │       ├── alert.tsx
│   │       ├── tabs.tsx
│   │       ├── pagination.tsx
│   │       └── ... (50+ UI components)
│   │
│   ├── lib/                       # Utilities and API
│   │   ├── api.ts                 # API client functions
│   │   │   ├── // Authentication APIs
│   │   │   ├── // Game APIs
│   │   │   ├── // Review APIs
│   │   │   ├── // User APIs
│   │   │   ├── // Admin APIs
│   │   │   └── // All endpoint functions
│   │   │
│   │   └── auth.tsx               # Auth context provider
```

```
|   |       ├── // AuthProvider component
|   |       ├── // useAuth hook
|   |       ├── // Login/Register/Logout logic
|   |       └── // Token management
|   |
|   ├── styles/                         # CSS modules
|   |   ├── components.css
|   |   ├── pages.css
|   |   └── utilities.css
|   |
|   └── guidelines/                     # Development guidelines
|
├── public/                             # Static assets
|   ├── images/
|   └── videos/
|
├── index.html                          # HTML template
├── vite.config.ts                      # Vite configuration
├── package.json                        # Node dependencies  ├── tsconfig.json                         #
TypeScript config
└── tailwind.config.js                  # Tailwind CSS config
```

9.2 Code Standards

Backend (PHP/Laravel):

- Classes: PascalCase (AuthController, User, Game)
- Methods: camelCase (register, updateProfile)
- Variables: camelCase ($user, $token)

- Database columns: snake_case (user_id, created_at) **Frontend**

(TypeScript/React):

- Components: PascalCase (GamesLibrary, Navbar)
- Functions/Variables: camelCase (fetchGames, userId)
- Interfaces: PascalCase (Game, User)
- Files: Component files PascalCase, utilities camelCase

9.3 API Endpoints Public Endpoints (No Authentication Required) Authentication

- POST /api/register - Register new user
- POST /api/login - User login

Games

- GET /api/v1/public/games - Get public games list

Protected Endpoints (Require Bearer Token) Authentication & Profile

- GET /api/user - Get current authenticated user
- GET /api/me - Get authenticated user data

POST /api/logout - Logout user

- PATCH /api/v1/user/profile - Update user profile **Users**
- GET /api/v1/users/{user} - Get user profile
- GET /api/v1/users/{user}/activity - Get user activity
- GET /api/v1/users/{user}/favorites - Get user's favorite games

Games

- GET /api/v1/games - Get all games (paginated)
- GET /api/v1/games/{game} - Get game details
- POST /api/v1/games/{game}/favorite - Add/remove game from favorites

Reviews

- GET /api/v1/reviews - Get all reviews
- GET /api/v1/reviews/{review} - Get specific review
- GET /api/v1/games/{game}/reviews/stats - Get game rating statistics
- POST /api/v1/games/{game}/reviews - Create review for a game
- PATCH /api/v1/reviews/{review} - Update review
- DELETE /api/v1/reviews/{review} - Delete review
- POST /api/v1/reviews/{review}/like - Like/unlike review
- POST /api/v1/reviews/{review}/comment - Add comment to review **Tips**

& Tricks

- GET /api/v1/tips - Get all tips
- GET /api/v1/tips/{tip} - Get specific tip
- POST /api/v1/games/{game}/tips - Create tip for a game
- PATCH /api/v1/tips/{tip} - Update tip
- DELETE /api/v1/tips/{tip} - Delete tip
- POST /api/v1/tips/{tip}/like - Like/unlike tip

POST /api/v1/tips/{tip}/comment - Add comment to tip **Wikis**

- GET /api/v1/wikis - Get all wikis

- GET /api/v1/wikis/{wiki} - Get specific wiki
- POST /api/v1/games/{game}/wikis - Create wiki for a game
- PATCH /api/v1/wikis/{wiki} - Update wiki
- DELETE /api/v1/wikis/{wiki} - Delete wiki
- POST /api/v1/wikis/{wiki}/like - Like/unlike wiki
- POST /api/v1/wikis/{wiki}/comment - Add comment to wiki

Comments

- GET /api/v1/comments - Get all comments
- GET /api/v1/comments/{comment} - Get specific comment
- PATCH /api/v1/comments/{comment} - Update comment
- DELETE /api/v1/comments/{comment} - Delete comment
- POST /api/v1/comments/{comment}/like - Like/unlike comment **Tags**
- GET /api/v1/tags - Get all tags
- GET /api/v1/tags/{tag} - Get specific tag
- GET /api/v1/tags/{tag}/games - Get games with specific tag **Leaderboard**
- GET /api/v1/leaderboard - Get leaderboard
- GET /api/v1/leaderboard/user/{user} - Get user rank

Achievements

- GET /api/v1/achievements - Get all achievements
- GET /api/v1/users/{user}/achievements - Get user achievements
- POST /api/v1/users/{user}/achievements/check - Check user

achievements POST /api/v1/users/{user}/achievements/{achievement}/unlock - Unlock achievement **Notifications**

- GET /api/v1/notifications - Get user notifications
- PATCH /api/v1/notifications/{id}/read - Mark notification as read
- PATCH /api/v1/notifications/read-all - Mark all notifications as read
- DELETE /api/v1/notifications/{id} - Delete notification **Posts**

- GET /api/v1/posts - Get all posts
- GET /api/v1/posts/{post} - Get specific post
- POST /api/v1/posts - Create new post
- DELETE /api/v1/posts/{post} - Delete post
- POST /api/v1/posts/{post}/like - Like/unlike post
- POST /api/v1/posts/{post}/comment - Add comment to post **Reports**
- POST /api/v1/reports - Submit report

10. References & Resources

List any resources, tutorials, or documentation used during development.

- **Laravel API Sanctum – For Security and Authentication**
  - **https://youtu.be/uyPvYnlzmhE**
- **UI and UX Designs**
  - **https://youtu.be/ODpB9-MCa5s**

Student/Team Signature: A.J. Jandrei N. Mercado **Date:** December 26, 2025