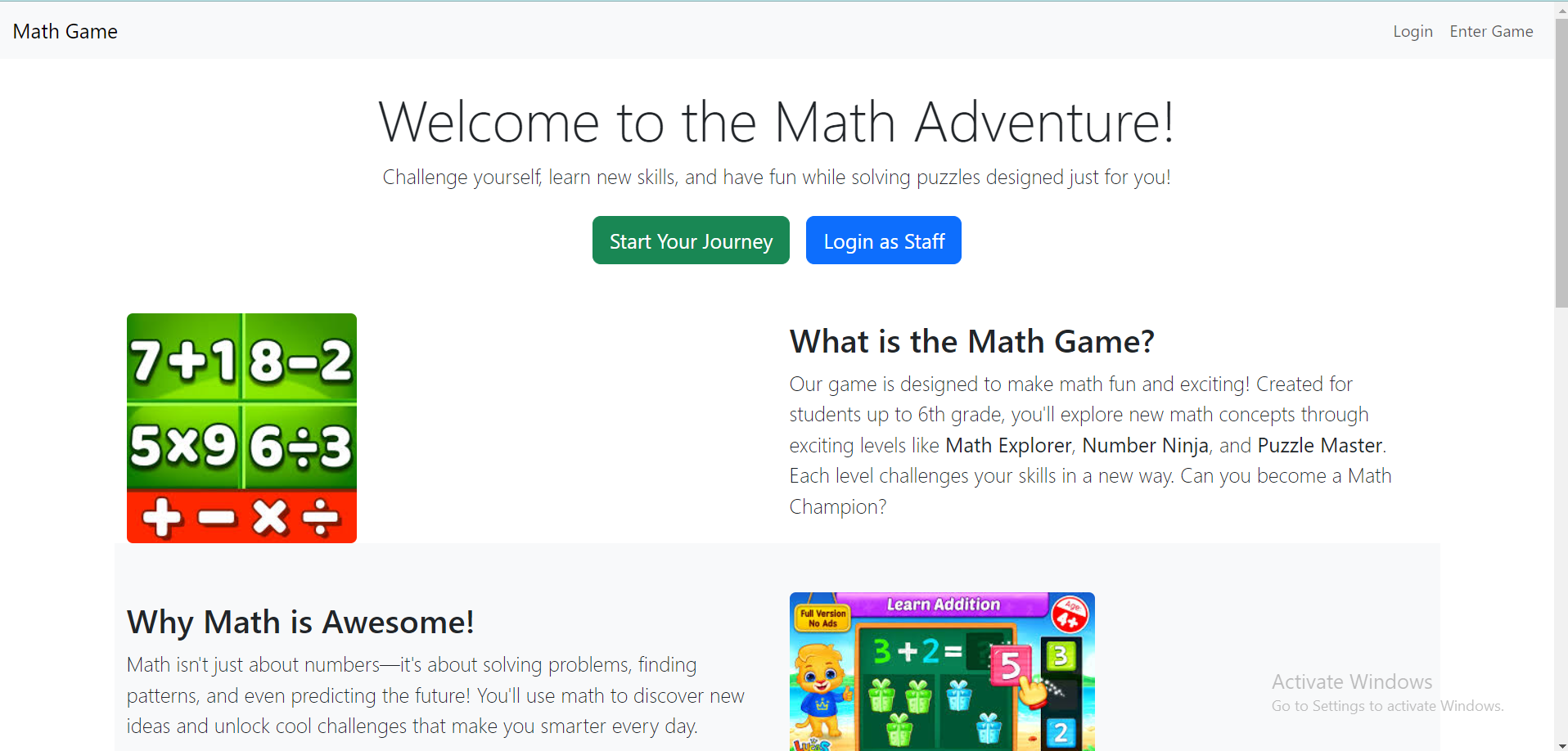
**Project Overview: Math Adventure Game (Django Application)**

**Introduction:**

This project is a Django-based web application designed for students to enhance their arithmetic skills through a level-based math game. It includes functionalities for both students and teachers, providing an engaging learning experience for students and a comprehensive progress-tracking dashboard for teachers.



**Application Features:**

1. **Home View (Landing Page):**
   * **Purpose**: Welcomes users and provides an overview of the Math Adventure game.
   * **Content**: The homepage contains a welcoming message, two main action buttons, and an explanation of the game.
     + **Buttons**:
       - **Start Your Journey**: Directs students to the game entry page.
       - **Login as Staff**: Takes staff members to the login page for staff access.
     + **Game Overview**: Explains the game levels such as *Math Explorer*, *Number Ninja*, and *Puzzle Master*, each focusing on different math concepts.
   * **Layout**: The homepage is visually engaging, featuring images and sections that highlight the game’s fun and educational aspects. The game is designed for students up to 6th grade.

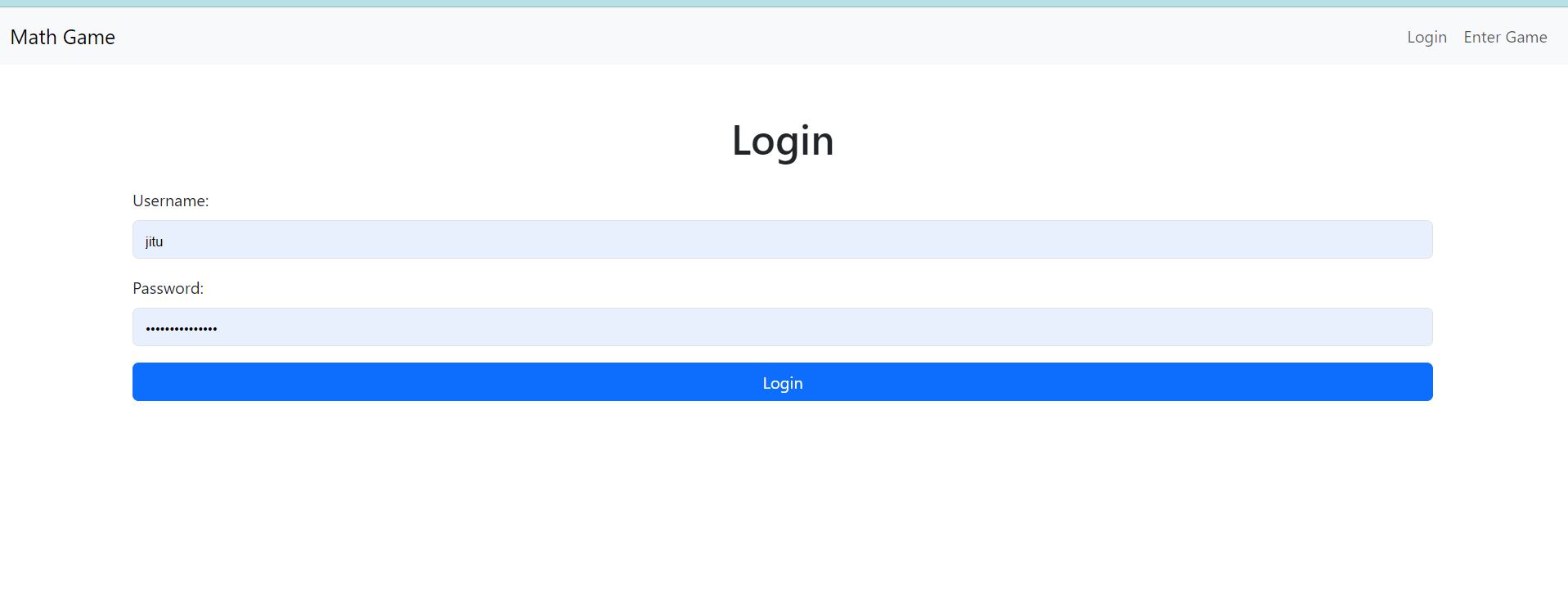
**Development**:

* + Created using Django’s template system, it extends a base template (base.html) to maintain consistency across all pages. Static files (like images) are served using Django's {% static %} tag for optimal loading.

1. **Login View (Staff Login):**
   * **Purpose**: Allows staff members (teachers or administrators) to log in and access the dashboard.
   * **Functionality**:
     + **Form**: A simple login form that asks for the username and password. It authenticates staff using Django's authentication framework.
     + **Access Control**: Only authenticated staff can access the teacher’s dashboard, while students are directed to the game pages.

**Development**:

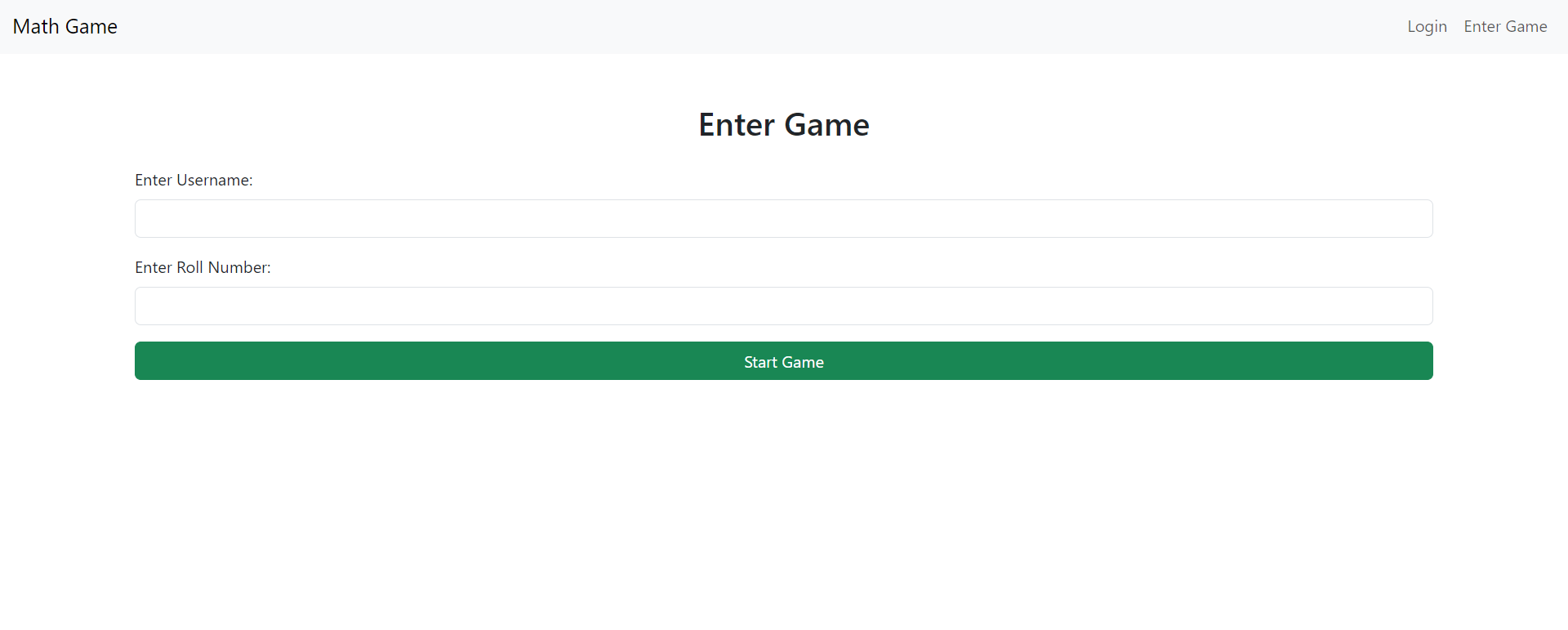
* + Utilizes Django’s built-in authentication system with custom user group checks to ensure only authorized staff can access sensitive information.



1. **Enter Game View (Student Entry):**
   * **Purpose**: Provides an interface for students to enter the game using their roll number and name.
   * **Functionality**:
     + **Form**: Students input their roll number and name to identify themselves. This data is stored and tracked in the system.
     + **Redirection**: After entering their details, students are redirected to the game’s level selection page based on their progress.

**Development**:

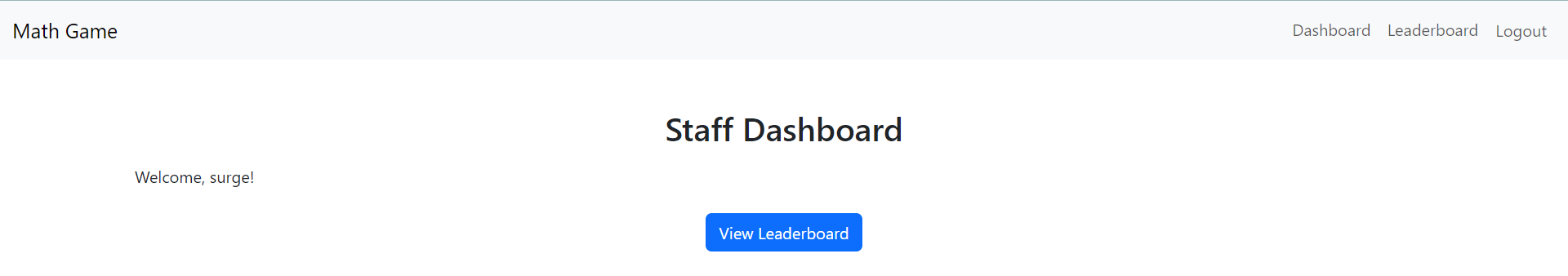
* + A simple form submission view that checks the student’s details and retrieves any progress stored in the database. If the student is new, they start from level 1.



1. **Dashboard View (Staff Progress Tracking):**
   * **Purpose**: Allows staff to track the progress of all students in the game.
   * **Content**:
     + **Student Progress Overview**: Displays a list of students along with their current level, the number of levels completed, and their overall score.
     + **Leaderboard**: An overview of top-performing students is available, providing teachers with insights into which students are excelling.
   * **Access Control**: This page is restricted to staff members only.

**Development**:

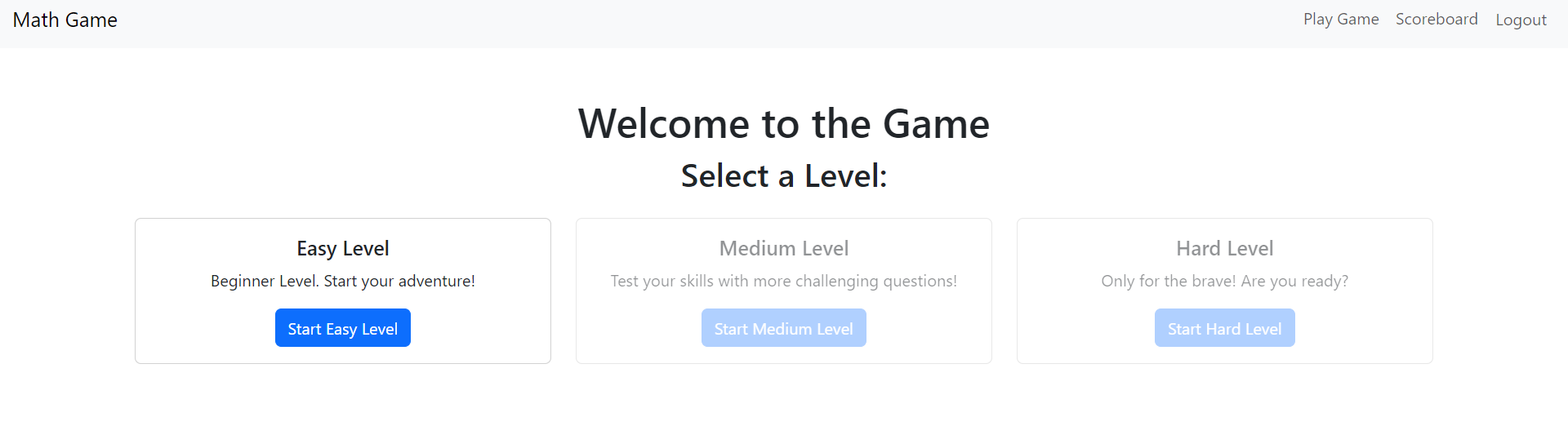
* + The dashboard view is built using Django’s query system to pull real-time data from the database, displaying each student’s progress and performance. Data is displayed in a structured table format for easy reading.

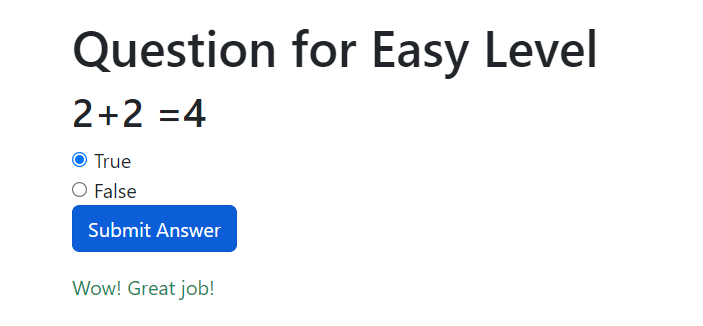


1. **Level View (Game Levels):**
   * **Purpose**: Allows students to access their current game level based on their progress.
   * **Functionality**:
     + **Automatic Progression**: Students are directed to the level they last completed. For example, if a student completed Level 3, they are automatically sent to Level 4 when they return.
     + **Problem Solving**: Each level contains math problems, ranging from basic arithmetic to more complex puzzles as students progress through levels like *Math Explorer* and *Puzzle Master*.
     + **Level Completion**: After completing a level, the system stores their progress and unlocks the next level.

**Development**:

* + The level view leverages Django’s session framework to store and retrieve student progress. Each level is designed with dynamic content generation based on the student's current level.





**Technical Stack:**

* **Frontend**:
  + HTML/CSS and Bootstrap for responsive design.
  + Template rendering is handled by Django’s template system, ensuring consistency across all views.
* **Backend**:
  + **Django Framework**: Handles routing, user authentication, and data storage.
  + **Django ORM**: Used to interact with the database to store student progress, staff details, and game levels.
  + **Session Management**: Student progress is tracked using Django’s session framework, ensuring a smooth user experience.
* **Database**:
  + A relational database (such as SQLite for development) is used to store data on students, their progress, and game details.

**How the Application is Built:**

1. **Project Setup**:
   * The project was initialized using Django’s standard startproject command, followed by creating an app named game\_app.
2. **Models**:
   * The Student model tracks each student’s name, roll number, and progress.
   * The Level model stores information about each game level, including the questions and answers.
3. **Views and URLs**:
   * Each view corresponds to a URL route, ensuring logical separation of concerns. For example, the home view is mapped to the root URL (/), while the login view is mapped to /login/.
   * Views for teachers (such as the dashboard) are protected with authentication and access control checks.
4. **Static Files**:
   * Images, CSS, and JavaScript files are served using Django’s static file handling system. The {% static %} template tag ensures these resources are loaded correctly across different environments.
5. **Templates**:
   * Templates are built with Django’s templating engine and structured to extend a common base template. This ensures consistency in the layout, header, and footer across the site.
6. **Authentication**:
   * Django’s built-in authentication system is used to manage staff login and restrict access to sensitive views such as the dashboard.
   * Staff members are assigned to a specific user group, ensuring that only authorized personnel can access the tracking features.

**Conclusion:**

This Math Adventure game provides an engaging platform for students to enhance their math skills, while offering teachers a powerful tool to track student progress and performance. By leveraging Django's robust framework, the project is built with scalability, security, and user experience in mind.