
Term Paper: ChessKid Connect - Facilitating Chess Partnerships

Author: Jing Bai

Date: 12/5/2023

Abstract

ChessKid Connect is a platform designed to address the challenges of finding suitable chess partners for players of various skill levels and preferences. This term paper outlines the motivation behind the project, the problems it aims to resolve, technical details of the platform, setup instructions, and future feature enhancements.

1. Motivation

The motivation behind ChessKid Connect stems from a real-world experience of a mother trying to find a suitable chess partner for her son. She explored several existing chess partner-finding platforms, including Chess.com and Reddit. However, she encountered two major issues:

1. **Outdated Partner Information:** Many platforms displayed partner information that was outdated, rendering the search results ineffective. Most individuals listed as potential partners were no longer interested in chess.
2. **Mismatched Chess Levels:** The platforms often failed to distinguish between players' chess levels accurately, resulting in mismatched partnerships. This mismatch led to unsatisfactory gaming experiences.

These challenges prompted the development of ChessKid Connect, a platform that aims to revolutionize the process of finding chess partners.

2. Problem Resolution

ChessKid Connect was conceived as a solution to the problems encountered in existing chess partner-finding platforms. The project aims to achieve the following:

- **Collect User Information:** Provide users with an intuitive and step-by-step wizard to input their relevant information, including name, contact details, age, chess level, preferred contact method, and gaming platform (online or local).
- **Customized Partner Matching:** Enable users to search for potential chess partners based on criteria such as chess level and gaming platform (online or local). This ensures that players are matched with compatible opponents.
- **Activeness Confirmation:** Periodically contact users to confirm their continued interest and activeness. This helps maintain a pool of active users for potential matches.

3. Technical Details

Platform Features

ChessKid Connect encompasses several key features:

- **User Info Wizard:** A user-friendly interface guiding users through the process of entering their information.

- **User Info Storage with NeDB:** Utilizes NeDB, a non-relational database, to securely store user data.
- **Partner Match Query:** Employs intelligent algorithms to match users based on criteria such as chess level and gaming platform.
- **Activeness Checking Email:** Sends periodic activeness confirmation emails to all users, ensuring a pool of active participants.
- **Donation Window:** Includes a donation section to support the platform's development and maintenance when a match result was found for a user.
- **Developer Portal:** A centralized hub for developers, housing three sub-pages: "All User Data View," "Match Result View," and "Email Sending Trigger View."

User Interface (UI) and User Experience (UX)

ChessKid Connect prioritizes a simple and predictable UI/UX design to enhance user engagement and satisfaction. The platform's interface is designed to be user-friendly, making it easy for users to navigate and input their information.

4. How to Setup and Run the Project

To set up and run the ChessKid Connect project, follow these steps:

1. Clone the Repository:

Clone the ChessKid Connect repository from the GitHub repository to your local machine.

```
git clone <git url>
```

2. Install Node.js:

Ensure you have Node.js installed on your system. You can download it from <https://nodejs.org/> if you haven't already.

3. Install Dependencies:

Navigate to the project directory and install the required Node.js packages using npm (Node Package Manager).

```
cd chesskid-connect
npm install
```

4. Setting Up Email Account (Optional):

To configure the email sending module, you'll need to provide your email account details.

- **Supported Email Services:** The project supports both Gmail and Hotmail accounts for email sending.
- **Gmail:** If you're using a Gmail account, please ensure that "Less secure apps" is enabled in your Google account settings. Note that using Gmail may require additional security setup.
- **Hotmail:** Hotmail accounts can be used without any additional security settings.
- Code Configuration

```
// Email Configuration (emailConfig.js)
```

```
const nodemailer = require('nodemailer');

const transporter = nodemailer.createTransport({
  service: 'Gmail', // Use 'Hotmail' if applicable
  auth: {
    user: 'your_email@gmail.com', // Replace with your email address
    pass: 'your_email_password', // Replace with your email password
  },
});

module.exports = transporter;
```

5. **Start the Application:**

Start the ChessKid Connect application by running the following command:

```
node app.js
```

The application will start, and you can access it by opening a web browser and navigating to <http://localhost:3000>.

6. **Access the Developer Portal:**

To access the Developer Portal, visit <http://localhost:3000/developer-portal>. Here, you can find links to the “All User Data,” “Match Result,” and “Send Email to All Users” pages for debugging and administrative purposes.

7. **Interact with the ChessKid Connect Platform:**

Use the platform’s features to input user information, find chess partners, and send activeness confirmation emails. Explore the various functionalities offered by the platform.

Conclusion

The ChessKid Connect platform offers a streamlined experience for finding chess partners and fosters a thriving chess community. Enjoy your experience and explore the potential for future enhancements and data analysis within the platform.