



Ollscoil
Teicneolaíochta
an Atlantaigh

Atlantic
Technological
University

B.Sc. (Hons) in Software Development Applied Project & Minor Dissertation

Project Proposal

Each student must complete the following form and submit it to their supervisor for consideration. Once your supervisor has signed-off on the proposal, you must submit the document for consideration using Moodle.

1. **Student Name:**
2. **Supervisor Name:**
3. **Project Name:**

4. **Project Context**

Describe the context of the problem domain here. Explain what you are proposing to do and your rationale for doing it. Explain why the problem domain is of interest.

Have you ever played chess before? ***Guardians of the Chess Grandmaster*** is designed to challenge and train players to think like a grandmaster. Through interactive puzzles, strategic missions, and real-time matches, users can develop tactical, positional, and endgame skills. These individuals have mastered the game over many years of study and practice, developing exceptional understanding and foresight. However, many aspiring players struggle to reach even a basic level of competence because they lack guidance, don't understand piece movement or strategies, or simply lose motivation along the way. This challenge highlights a need for an accessible and engaging way to learn chess - one that helps players not only understand the game but also appreciate its depth and artistry.

To address this issue, the proposed solution is to develop an **interactive digital learning platform** that simplifies the process of learning chess through guided tutorials, visual aids, and engaging practice sessions. The platform will combine **user-friendly design, gamification, and educational modules** to make chess learning both fun and effective. Beginners will be able to progress at their own pace, mastering basic moves, tactics, and strategies before advancing to more complex challenges. This approach aims to remove the intimidation often associated with learning chess, making it more accessible to all users regardless of age or experience.

This problem domain is particularly compelling because it connects **education, technology, and cognitive development**. Chess has long been recognized for enhancing memory, logical reasoning, and strategic thinking. Yet, its complexity often discourages newcomers from engaging with it. By leveraging technology to lower the learning barrier, this project has the potential to make chess more inclusive, spark global interest, and contribute to intellectual growth across diverse communities and age groups. Ultimately, it promotes not only the love of the game but also the development of valuable life skills through interactive learning.



5. Project Objectives

Write out the key objectives of the project as bullet points. Each objective should be clear, realisable and measurable / testable, i.e. the success of your project is determined by the degree to which these are realised.

- Design and develop a user-friendly **Landing** interface that includes *Login & Sign Up* options. **Measurable:** *successful user authentication and password recovery tests.*
- When your logging in to your account, create a functional and interactive **Home** page that provides access to all main features such as game modes, tutorials, friends and user profile on the bottom navigation, especially your Game History (E.g. accuracy moves, opponent name). **Testable:** *verifying navigation links, responsiveness and checking out your history data.*
- Develop a user **Profile** page where players can view and edit personal information (Name, Email & Rating), game history, and **Settings** logo (Navigate to Settings). **Testable:** *ensuring profile updates are saved and displayed correctly.*
- Include a **Settings** section that allows customization of preferences (e.g., theme, theme of the chess board, set of chess pieces, clear cache, install PWA, logging out). **Measurable:** *verifying that user preferences persist after restarting the app.*
- Provide an in-app **Tutorial** section that teaches users how to play chess, covering rules, piece movements, and strategies. **Testable:** *by ensuring tutorial content loads correctly and is accessible to new users.*
- Implement a **Play** feature allowing users to choose between Player 1 vs Player 2 (NO Difficulty Level) enter your name & other name, or Player vs AI modes (Difficulty Level) without enter your name, either with or without a timer. **Testable:** *verifying both modes function properly and the AI responds as expected, identify their names and what rating were they on.*
- Develop a **Friends** feature that enables users to add, view, search, and challenge friends within the app. **Testable:** *confirming friend requests, acceptance, and in-game invitations work properly.*
- Implement a secure **Logout** functionality that safely ends the user session and redirects to the landing page. **Testable:** *ensuring the user cannot access protected pages after logging out.*



6. Technologies & System Architecture (Part 1)

Explain the technologies you are going to use and why you selected them. These include the programming languages, operating systems, presentation and storage technologies and any cloud / 3rd party libraries / services that you intend to use.

- **Frontend:** **React** will be used with **Vite** and **TypeScript** to build a dynamic, component-based interface. Vite provides a fast development environment and optimized build process, while React's virtual DOM ensures efficient updates and smooth integration with backend APIs. The presentation layer will combine React components with **TypeScript**, **HTML**, **CSS**, and *minimal JavaScript* to deliver a responsive, browser-based interface.
- **Backend:** the project will use **MongoDB** in combination with **Node.js** and **Express.js**. This setup provides full flexibility to define server-side logic, manage user authentication, and store game-related data efficiently. E.g. change index.js to index.ts.(TypeScript)
- **Database:** **MongoDB** will be used as a NoSQL document-based database to store user profiles, match histories, move data, ranking statistics, add them to your **Friends** page. Its flexible, schema-less structure makes it ideal for handling dynamic data such as live game states and player progress. **Mongoose** will be integrated as an Object Data Modelling (ODM) library, providing schema definitions, validation, and streamlined database interactions. **Express.js** can handle API routes and manage communication between the *Frontend (React)* and the MongoDB database. **Deployment** could be hosted on MongoDB Atlas and deployed using Render, Railway, or Vercel (Serverless Functions) for scalable and efficient hosting.
- **OS:** The project is **web-based**, so it is compatible with **any OS** that supports a modern web browser, including **Windows**, **macOS**, and **Linux** by using Progressive Web Apps (PWAs) for example. All development tools, such as **Node.js**, **Vite**, and **React**, are **cross-platform**, ensuring flexibility for development, testing, and deployment.
- **Third-Party Libraries and Services:** The project will integrate few third-party libraries to simplify development and add advanced functionality: (Next Page on Part 2)



7. Technologies & System Architecture (Part 2)

Library / Service	Purpose	Reason for Selection
Types/chess.ts	Game logic and rules of chess	Develop logical thinking and problem-solving skills through implementing the rules and strategies of chess
Utils/chessLogic.ts	Game logic and move validation	Provides a complete chess rules engine to handle valid moves, checkmates, and draws.
Utils/chessAI.ts	Player AI (Computer or Robot)	Provides AI to make a valid moves, checkmates, and draws
framer-motion	Animations and transitions	Adds smooth, visually appealing animations for piece movement and UI effects. E.g. Use it as a Slide Show
react-router-dom	Page routing and navigation	Enables seamless navigation between app sections (Home, Play, Profile, etc.).
Firebase Auth	User login and registration	Secure and easy authentication management.



Ollscoil
Teicneolaíochta
an Atlantaigh

Atlantic
Technological
University

8. Schedule of Work

Using a Gantt chart or tabular format, outline your schedule of work for all the key project activities, deliverables and dates.

Task List	Start of Date	End of Date
Project Proposal (Project Definition, Research Requirements & Gathering)	15/9/2025	31/10/2025
System Architecture Planning (E.g. link on proto.io, research, AI Model, Integration)	1/10/2025	5/12/2025
Frontend Development (React)	1/11/2025	14/2/2026
Backend Development (Node.js & Server.js)	1/11/2025	28/2/2026
Database Setup (MongoDB)	1/12/2025	31/1/2026
AI Model Development (Data Collection, Training, and Optimization)	19/1/2026	15/3/2026
Integration (Frontend + Backend + Model)	1/2/2026	31/3/2026
Testing & Quality Assurance	1/11/2025	30/4/2026
Project Documentation & Dissertation	1/1/2026	30/4/2026
Final Presentation & Submission	1/4/2026	30/4/2026