## **TЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

**по професия код 481020 „Системен програмист“**

**специалност код 4810201**  **„Системно програмиране“**

Тема: Система за провеждане на игра на шах

между човек и алгоритъм

Дипломант: Дипломен ръководител:

*Кристиян Михайлов Петров Николай Станишев*

СОФИЯ

2 0 2 2

# Използвани термини

**WEB** - The World Wide Web – Световната мрежа

**API** - Application Programming Interface – Интерфейс за програмиране на приложения

**SQL** - Structured Query Language – Език за съхранение, манипулация и получаване на данни от таблици

**NOSQL** - Non-Structured Query Language – Нерелационен и неструктуриран език за обработка на данни.

# Увод

Ежедневно напредва развитието в областта на информационните и комуникационни технологии. Благодарение на него популяризирането и употребата на Интернет пространството се увеличава.

В края на 2015г. в световен мащаб активните потребители на Интернет са около 3млрд., докато през 2021г. тази бройка расте до близо 5млрд.[1] Хората са си изградили ежедневен навик, който представлява използване на десетки сайтове за добиване на информация, без да се замислят за процеса, който стои зад всичко това. Много от потребителите не осъзнават какво всъщност достъпват и по какво се различава от останалите. WEB сайтът не позволява възможност за интеракция от страна на своите потребители, тоест той просто им предоставя информация, картинки, видеа и аудио, те са статични и не се обновяват динамично. WEB приложенията са изключително функционални и интерактивни, в повечето случаи очакват някакво действие от потребителя и обновяват своята информация динамично, като доста често са свързани с база данни, където съхраняват необходимите за тях неща. [2]

С раждането на толкова много възможности за създаването на креативни идеи и употребата им от други потребители, се появяват и игрите. През 2021г. бройката на хората, които играят видео игри е около 3млрд. [3] Това е една много популярна и бързо нарастваща област, която намира все повече последователи от всички възрасти.

С настоящата дипломна работа ще бъде изградена система за WEB базирано приложение за игра на шах. За реализацията на тази идея, тя може да бъде разделена на 3 основни части.

Първата част представлява графичния потребителски интерфейс, който визуално представя на потребителя шахматна дъска и фигурите. Играчът има възможността да мести пионките си с натискането на бутон на мишката. WEB приложението съдържа няколко страници. Начална, от която е възможно да се достъпят другите две. Страница за влизане в стая, в която трябва се вписва даден код, за присъединяване към чужда игра. Третата страница е самата игра на шах, в която се играе играта, дава възможност за копиране кода на стаята с натискането на бутон или обновяване на дъската с начално наредени фигури.

Втората част е създаването на алгоритми, които проверяват изиграните от играча ходове, вярността и възможността им спрямо правилата на играта шах. Обновяване на информацията за шахматната дъска, която бива променена от потребителския интерфейс или базата данни, чрез изпращане на заявки, към създаденият API, за запаметяване или взимане на данни.

Третата част е свързването на графичния потребителски интерфейс, WEB сървъра и базата данни. Създаването на API, който позволява манипулирането на данни свързани с играта на всеки играч. Поддържането на различни адреси, от които могат да се достъпят различните страници на WEB приложението. Отваряне на WebSocket канал, чрез който играчите в дадена игра да сигнализират помежду си за извършения от тях ход.

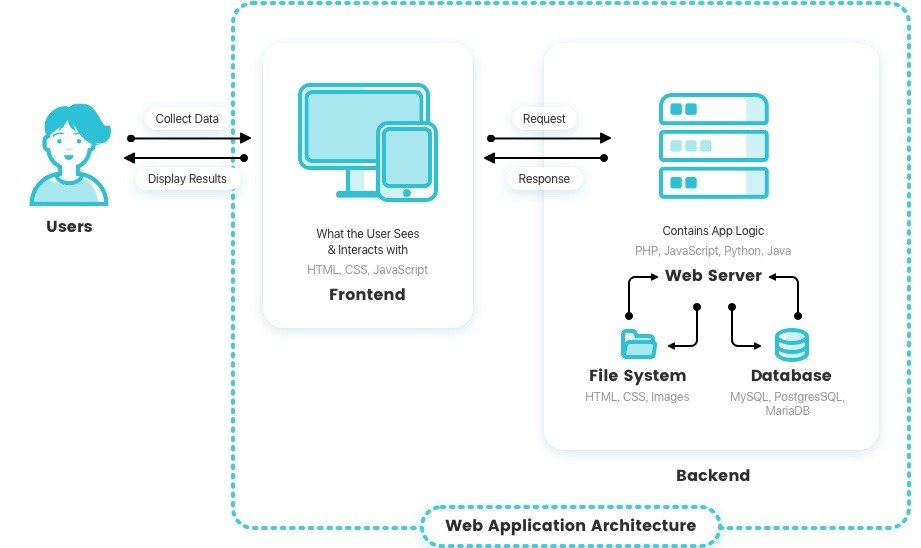
Проблемът, който решава системата е улеснена и приятна за използване платформа за възпроизвеждане на игра на шах между двама опоненти, която не изисква никаква аутентикация.

# ПЪРВА ГЛАВА

*МЕТОДИ И ТЕХНОЛОГИИ ЗА РЕАЛИЗИРАНЕ НА WEB БАЗИРАНО ПРИЛОЖЕНИЕ ЗА ИГРА НА ШАХ*

## 1.1 Основни принципи, технологии и развойни среди за разработване на WEB приложение

### 1.1.1 Обзор на WEB приложение

За създаването на едно WEB приложение са необходими контролер и графичен потребителски интерфейс(ГПИ). В най-честия случай, потребителят изпраща заявка към контролера, през ГПИ, която се обработва и извлича нужната информация от базата данни, след което данните се изпращат обратно и се изобразяват пред потребителя. Доста често е необходимо да се въведе и WebSocket връзка, която представлява отваряне на канал, чрез който сървърът комуникира с един или няколко клиента едновременно.

***Фиг. 1.1*** *Архитектура на WEB приложение*

### 1.1.2 Принципи за разработване на WEB приложение

Разработката на проект е дълъг и сложен, изисква доста стъпки, за да се получи качествен продукт. [4]

* **Създаване на идея** - Идеята трябва да бъде породена от проблем, който ще бъде решен от продукта.
* **Съществуващи системи** - Трябва да бъде уникално и различимо от другите решения.
* **Технологии** - Избор на релевантни технологии за създаване на WEB приложение, съобразно идеята и тяхната поддръжка на библиотеки и и използваемост.
* **Планиране на основата** - Планирането на основните функционалности на проекта е важно, за да се създаде приоритизация.
* **Менажиране** - Използване на среда за менажиране на апликацията, в която се следи прогрес, приоритетни задачи и т.н
* **Скициране на дизайн** - Важно е да съществува ясна представа за това как ше изглежда приложението, за да бъде разработено лесно след това.
* **База данни** - В повечето случаи необходима, за запазване на данни, които се използват за визуализация или съхраняване на важна информация използвана от алгоритми.
* **Контролер** - Той най-често съдържа базата данни, сървъра, логиката и методи за обработка на заявки.
* **ГПИ** - Създаване на графичен потребителски интерфейс, на база скиците, изчистен, лесен за употреба и интерактивен. По възможност поддържащ мобилни устройства.
* **API-ГПИ** - Необходима връзка, чрез която се визуализират необходимите от потребителя данни, чрез заявки към базата данни.
* **Интернет** - За да бъде използван проекта от други хора, трябва да бъде качен в интернет пространството, като бива пуснат на машина поддържаща го работещ.

### 1.1.3 Технологии за разработване на WEB приложение

Избирането на технологиите за разработка на WEB приложение е важен процес, който трябва да бъде съобразен с идеята и скалируемостта на проекта. Те могат да бъдат разделени в няколко подкатегории:

* **Технологии за създаване на ГПИ** - Това са тези технологии, които са използвани за направата на красив и удобен графичен потребителски интерфейс, чрез който потребителите да управляват приложението. Такива са “HTML” (Език за създаване на структурата на една страница), “CSS” (Език за украсяване на визуално предстваените елементи в страница), “JavaScript” (Език използван за добавяне на допълнителна функционалност на ГПИ).
* **Технологии за създаване на контролер** - Това са програмни езици, които позволяват създаването на сървър, приемане на заявки, управление на базата данни и поддръжката на библиотека за разширяване на функционалността. Често използвани програмни езици са “Python”, “Java”, “Javascript”, “PHP” и т.н.
* **Технологии за съзадаването на база данни** - Базата данни е място за съхранение на важна, ключова информация. Съществуват две основни разновидности “SQL” и “NOSQL”. Важните разлики са, чe SQL базите данни са релационни и структурирани, използващи таблици за запазване на информация. При NOSQL не се използват таблици, а ключови стойности, графи и т.н. Те нямат определена структура и не са релационни. Най-разпространени за SQL са “Db2”, “MySQL”, “PostgreSQL”, “SQLite”, “Microsoft SQL Server” и т.н., а за NOSQL са “Redis”, “FaunaDB”, “MongoDB”, “Cassandra” и други.
* **Framework** - Това е вече изготвена основа за изграждане на WEB приложение. Употребата им е широко разпространена, защото улесняват процеса, понеже не се изготвя апликацията от нищото, ами вече има изградена структура. Съществува голяма разновидност, според избрания програмен език. Примери са “Django”, “Express.js”, “Vue.js”, “Laravel”, “Angular”, “Flask” и т.н.

### 1.1.4 Развойни среди за разработване на WEB приложение

**1.2 Алгоритъм за описване на достигната позиция позиция повреме на игра на шах (FEN)**

Нотацията на Форсайт-Едуардс или FEN е стандартна нотация за описване на позициите на фигурите по време на игра на шах.

FEN е важен и необходим на едно приложение за игра на шах, защото улеснява пренасянето на състоянието на дъската, като го превръща в един ред. Използва се при запаметяване и вземане на информация, като използва 8 части отделени със символа “/”, които представят един ред от дъската. Всяка една част съдържа съкратена версия на наименованията на фигурите и празните полета на този ред. Буквите, които използва репрезентират фигурите на дъската:

- “p” е пешка

- “r” е топ

- “n” е кон

- “b” е офицер

- “q” е царица

- “k” е цар

Разграничаването на това дали дадена фигура е черна или бяла става с използване на главната буква на дадената пионка, това показва, че тя е от бял цвят. Празните полета на реда се представят с число от 1 до 8, в зависимост от това колко последователни празни полета има.

**Фигура 1.1**

Тук подредбата е:

“rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR”

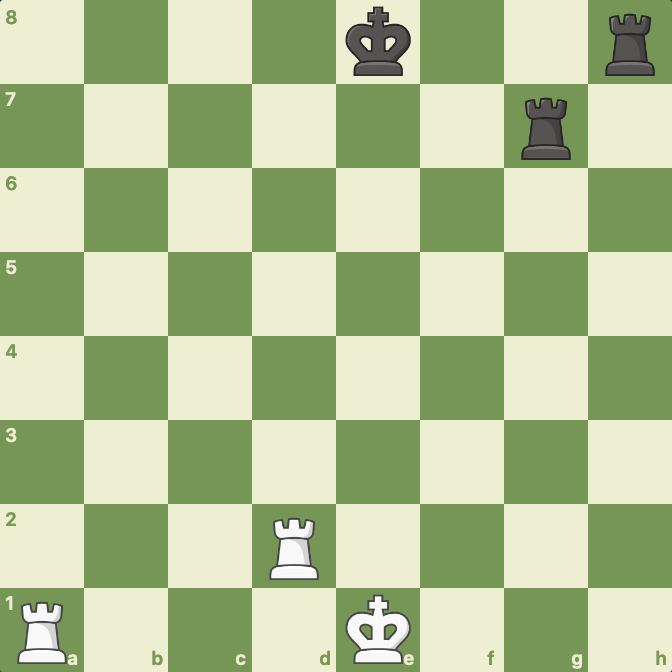
**Фигу****ра 1.2**

Тук подредбата е:

“r1b1k1nr/p2p1pNp/n2B4/1p1NP2P/6P1/3P1Q2/P1P1K3/q5b1”



Нотацията също показва кой цвят е на ход. След подредбата на фигурите се изписва буквата “w”, която показва ще белите фигури трябва да местят или “b” аналогично за черните фигури.

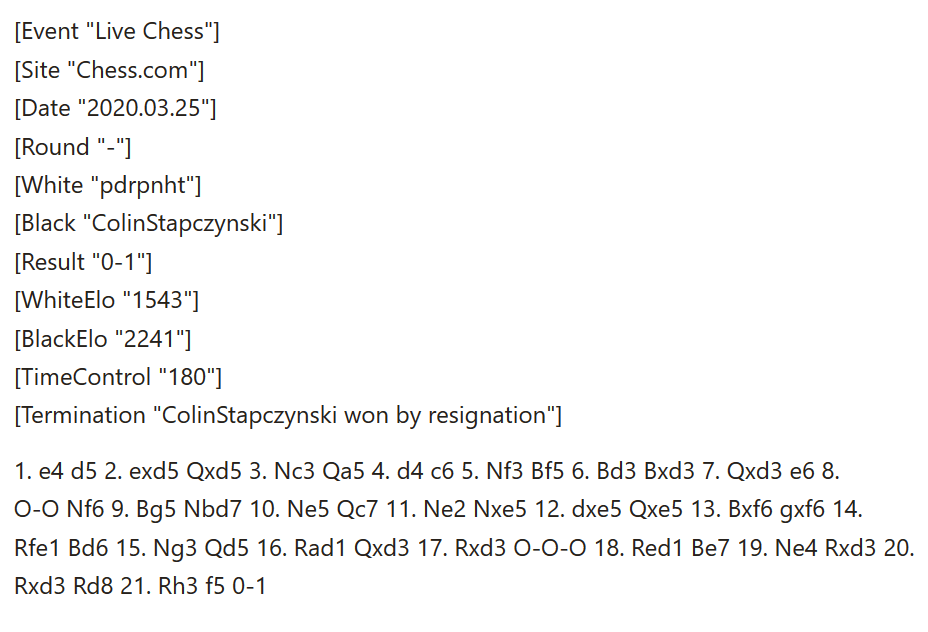
Възможността за рокада също се включва. Буквата “q” се използва за рокада от страната на царицата, а буквата “k” показва, че е възможна рокада от страната на царя. Когато буквите са големи, рокадата са отнася за белите фигури, а малките букви, за черните. При използване на символа “-” се показва, че няма възможна рокада за нито една от страните.

**Фигура 1.3**

“FEN: 4k2r/6r1/8/8/8/8/3R4/R3K3 w Qk”

**1.3 Алгоритъм за записване на всички изиграни ходове в една шахматна игра (PGN)**

Нотацията за преносима игра е стандартен формат за записване на игра под формата на текст. Тя е създадена от Стивън Едуардс и подпомага зареждането на шахматната игра от компютри.

Тя е по-различна от **FEN (1.2)**, тъй като тя записва поредицата от ходове направени по време на игра, а не просто състоянието на шахматната дъска. Тази нотацията също записва подробна информация, като например имената на играчите, таймера, резултата и още много.

**Фигура 1.4**

Това е информацията, която нотацията запазва

**Глава Втора**

**ПРОЕКТИРАНЕ НА СТРУКТУРАТА**

**НА WEB БАЗИРАНО**

**ПРИЛОЖЕНИЕ ЗА ИГРА НА ШАХ**

**2.1 Функционални изисквания на проекта**

Проектът се състои от три основни компонента WEB сървър, API и графичен потребителски интерфейс. Приложението трябва да представлява система, която позволява на потребителя да играе шах, като се спазват правилата на играта. Тази система ще представлява интернет приложение, което предоставя приятен и лесен за използване графичен интерфейс. Потребителят ще мести фигурите, които ще бъдат обработвани от логиката на приложението и ще бъдат запазвани от базата с данни. Към неговата игра ще може да се свързва друг потребител, с когото да играят. Използва се WebSocket комуникация, за обновяването на шахматната дъска при местене на фигурите.

**2.1.1 WEB сървър**

WEB сървърът ще зарежда на определени адреси графичния интерфейс и API, които могат да бъдат достъпвани и манипулирани. Той също ще трябва да управлява съхранената информация. Благодарение на API, той взима информацията въведена или поискана и контролира базата с данни. Той трябва да и да позволи свързването чрез WebSocket комуникация за двамата потребители, които ще играят.

**2.1.2 API**

API трябва да поддържа заявки за промяна на състоянието на шахматната дъска и дали трябва да се мести бяла или черна фигура. Трябва да съдържа начин за свързване на дадена дъска с потребител и да може да се взима огранича информация за играта.

**2.1.3 Графичен потребителски интерфейс**

Улесенен и приятен за използване графичен интерфейс, който да визуализира шахматната дъска и нейните фигури. Да позволява на потребителя да мести фигурите и да проверява валидността на дадения ход. Трябва да може да прави заявки, чрез които да обновява състоянието на дъската или да взима запазеното.

**2.2 Среда за разработка, езици и библиотеки**

**2.2.1 WEB сървър**

Езикът избран за реализацията на WEB сървъра е **Python**, а по-точно framework **Django**.

**2.2.1.1 Python**

**Python** е програмен език от високо ниво, той няма конкретна употреба, понеже се използва за всичко. Той е един от най-използваните езици и то поради големия набор от библиотеки, лесната му четимост и мащабното му приложение

**2.2.1.2 Django**

**Django е** WEB framework за **Python**, който позволява бързо и лесно създаване на нов проект. Използва улеснен начин за манипулиране на база с данни (“Model”), система за обработка на HTTP заявки (“View”) и още много функции предназначени за WEB приложения

**2.2.1.3 Pipеnv**

При разработката на това приложение е необходимо инсталирането на различни библиотеки за **Python**. **Pipenv** е съвкупност от мениджър на библиотеки и виртуална среда, като той предоставя улеснен начин за инсталацията на пакети във виртуална среда. Виртуалната среда е необходима на едно WEB приложение, за да може инсталацията на странични библиотеки да бъде само за този проект и да не се смесва с други приложения.

**2.2.1.4 SQLite3**

Почти всички WEB приложения използват база с данни. Това е място за съхранение на данни свързани с апликацията. За реализацията на този проект е необходима таква, за запазването на информация свързана с шахматната дъска и нейните потребители. При създаването на **Django** проект се създава лесна за използване база с данни **SQLite3**.

**2.2.1.5 Django REST framework**

Това е библиотека за създаване на WEB API. Тя има много добра интеграция с **Django**. Поддържа “Serialization”, което представлява превръщането на “Django Model” в Python тип данни, която могат да бъдат използвани, като JSON, XML.

**2.2.2 Графичен потребителски интерфейс**

За направата на графичния потребителски интерфейс е избран програмния език **JavaScript**, а по-конкретно библиотеката **React**.

**2.2.2.1 JavaScript**

**JavaScript** е програмен език, който поддържа обектно ориентиран и функционален стил на програмиране. Той е един от най-популярните и използвани езици. Най-често се използва в WEB приложенията, за добавяне на допълнителна функционалност.

**2.2.2.2 React**

Това е библиотека на **JavaScript** за създаване на потребителски интерфейси. Той поддържа компоненти, които могат да менажират своето състояние и да се презареждат автоматично. Това е една от най-разпространените библиотеки на **JavaScript** при създаването на потребителски интерфейси и има широк набор от библиотеки направени специфично за съвместна употреба с нея, разширявайки основните ѝ функционалности.

**2.2.2.3 Babel**

**Babel** е компилатор за **JavaScript**, който позволява превръщането на ECMAScript 2015+ код в използваем такъв за стари версии на **JavaScript** в браузърите.

**2.2.2.4 React-Bootstrap**

Това е най-разпространеният framework за **React**. Той е създаден за улеснена разработка на графичния потребителски интерфейс. В него се съдържат вече готови и оптимизирани елементи, които могат да се приложат в WEB приложението. Той позволява да се промени стила, размера и други функционалности на елементите, така че да приложението да бъде красиво и персонализирано.

**2.2.2.5 React-Icons**

**React-Icons** е една от най-популярните библиотеки за икони в **React**. Тя съдържа голям избор от иконки, които може да използваме.

**2.2.2.6 ReactDOM**

Той служи за използване на специфични DOM функции, които могат да се използват в апликацията.

**2.2.2.7 React Router DOM**

Тази библиотека предоставя възможността за създаване на маршрутизатор на страници.

**2.2.2.8 Webpack**

**Webpack** представлява библиотека, която конфигурира **JavaScript**, така че да пакетира кодовете. Тази библиотека предоставя възможността за режим на разработка на проекта, който е оптимизиран, работи по-бързо и ни връща подробни съобщение за грешки.

**2.2.2.9 React-copy-to-clipboard**

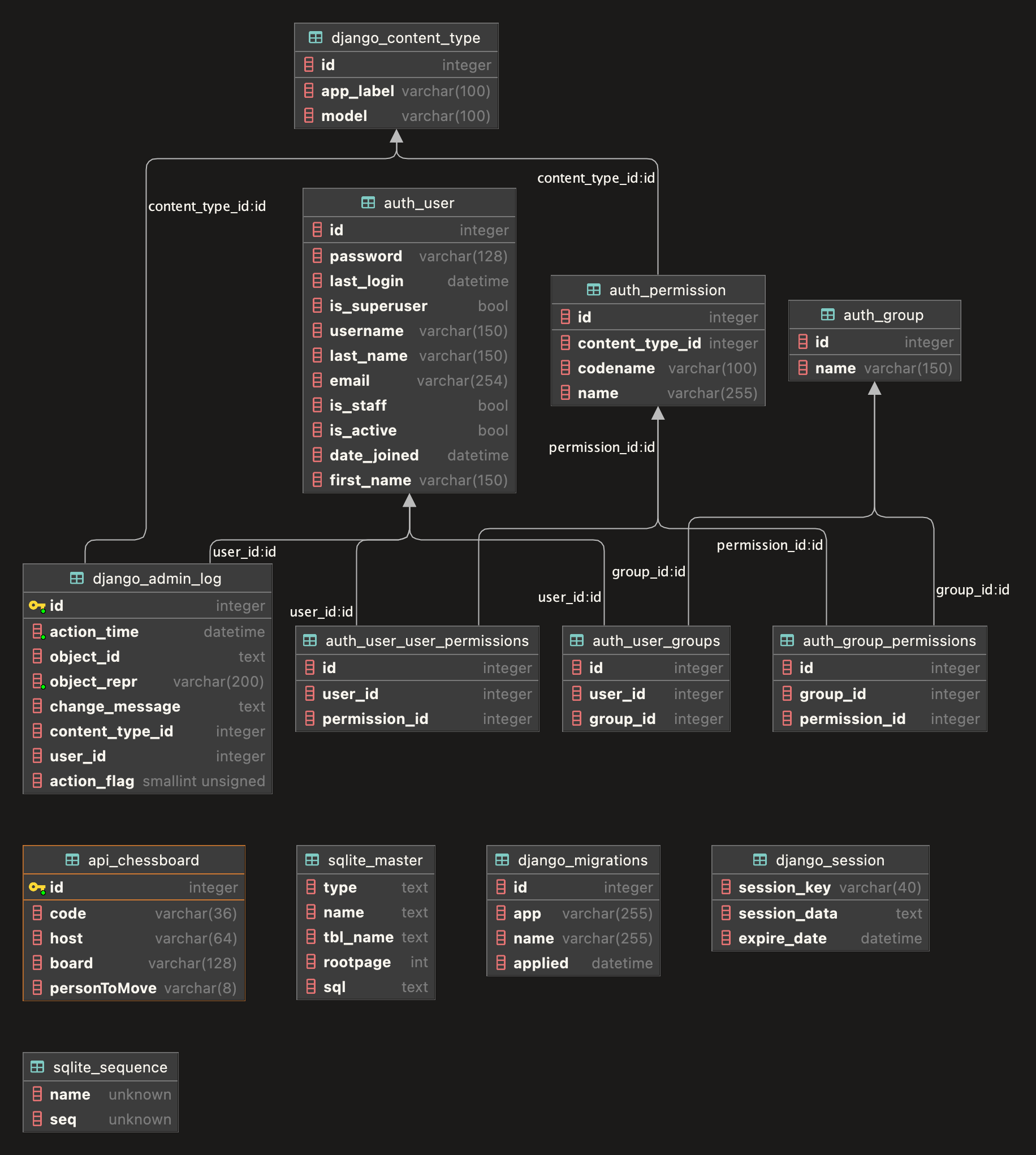
**React-copy-to-clipboard** е библиотека, която ни дава възможността за създаване на **React** компонент, който копира избраната стойност и я поставя в клипборда на потребителя.

**2.2.2.10 React-animate-text**

Това е библиотека, която ни предоставя вече готов **React** компонент, който променя информацията в него, като я показва с анимация тип пишеща машина.

**2.2.3 Среда за разработка**

Проектът е написан с помощта на **PyCharm**. Това е IDE използвано за компютърно програмиране, направено от Чешката компания JetBrains. То е предназначено за езика **Python**, но поддържа и други. Има приятен и красив интерфейс, който може да бъде променян спрямо нуждите на потребителя. Има много приставки, който улесняват и забързват процеса на писане на код. Създаването на **Django** сървър и конфигурирането му е улеснено.

**2.3 Структура на базата с данни**

**Фигура 2.1**

Диаграма на базата с данни

**Трета глава**

**3. Програмна реализация на WEB базирано приложение за игра на шах**

# Използвана литература

1 - Number of internet users worldwide from 2005 to 2021

<https://www.statista.com/statistics/617136/digital-population-worldwide/>

2 - Website vs Web App: What’s the Difference?

[https://medium.com/@essentialdesign/website-vs-web-app-whats-the-difference-e499b18b60b4/](https://medium.com/@essentialdesign/website-vs-web-app-whats-the-difference-e499b18b60b4)

3 - Number of video gamers worldwide in 2021

<https://www.statista.com/statistics/293304/number-video-gamers/>

4 - A 15-Step Guide on How to Build a Web App

[https://www.netsolutions.com/insights/how-to-build-a-web-app/#1-source-an-idea/](https://www.netsolutions.com/insights/how-to-build-a-web-app/#1-source-an-idea)