**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Сайт за игра на шах

Дипломант: Научен ръководител:

*Благовест Атанасов Таня Петрова от Astea Solutions*

СОФИЯ

2 0 2 1

**WEB приложение за игра на шах**

**УВОД**

Шахът – една от най-известните логически игри и единствената настолна игра, която се зачита за спорт. Шахът винаги е бил модерен, като правилата му не са сложни, но изисква много мислене и правилно преценени ходове.

Реших да имплементирам играта на шах защото ме заинтригува идеята да измисля реализацията на тази логическа главоблъсканица във формата на код. Играта е толкова популярна, че има десетки разработени и публични имплементации, което изглежда сякаш обезсмисля моята идея. Аз обаче реших да направя своята реализация на проекта без да взимам никакъв готов код или алгоритми, да се потопя във всеки аспект на играта и с цената на повече време да успея да се справя сам. Успях да имплементирам играта, която стана ядрото на цялата дипломна работа, и макар да не е нищо ново или оригинално, за мен е ценен опит и чудесен първи голям проект.

Изискванията за дипломната работа са следните:

- Поддръжка на потребители

- Играене на шах в мултиплеър

- История на изиграните игри

- Рейтинг система на играчите

- Поддръжка на турнири

- Чат между играчите по време на играта

**ПЪРВА ГЛАВА**

**МЕТОДИ И ТЕХНОЛОГИИ ЗА РЕАЛИЗИРАНЕ НА WEB ПРИЛОЖЕНИЯ**

* 1. **Основни принципи, технологии и развойни среди за реализиране на WEB приложения**

WEB приложенията представляват софтуер, в който са съчетани две части – сървър и клиент.

Сървъра представлява мястото, където се съхраняват, изпращат и извикват данни, като неговата функционалност позволява на клиентите да разполагат с конкретни услуги.

Страната на клиента представлява тази част от софтуера, която е ориентирана към крайния потребител. От тук се извикват заявки към сървъра и получените данни биват обработени и показани на потребителя по възможно най-удобния и практичен начин.

Сървърът най-често се състои от две части – приложение и база данни. Приложението може да бъде написано на голям набор от езици за програмиране, като най-използваните са PHP, Python и Java. Базата данни може да бъде създадена с различни СУБД, като най-известните са MySQL, SQL Server и PostgreSQL. Базата данни съхранява информацията, която е нужна в таблици, а чрез приложението тази информация бива създадена, редактирана и извиквана от клиента за по-нататъшна обработка.

Страната на клиента представлява WEB частта на приложението, на практика уебсайта. Той е изграден чрез сбор от HTML, CSS и JavaScript, като HTML представлява структурата на страницата, CSS оформя дизайна на елементите на базово ниво, а JavaScript е езика,

който обработва заявките към сървъра и прави страницата функционална и представителна за потребителя.

Съществуват много библиотеки и разширения както за сървърната, така и за клиентската част от приложението. Отдолу са показани най-известните от тях.

За сървърната част:

* PHP: Laravel, Codelgniter
* Python: Django, Flask
* Java: Spring, Struts

За клиентската част (JavaScript):

* Angular
* React
* Vue

Използвайки библиотеки и разширения програмистите улесняват и забързват процеса на работата си, защото с по-малко написан код свършват повече работа и съответно проектите могат да бъдат развивани до по-голяма степен на сложност много по-ефективно.

* 1. **Съществуващи решения и реализации – chess.com & lichess.org**

****

****

Когато говорим за WEB приложение за игра на шах, съществуват много реализации на такава платформа, като ще разгледаме двете най-използвани – “chess.com” и “lichess.org”. Двете платформи имат много сходни характеристики и функционалности, но все пак имат някои определени разлики.

Сайтът “chess.com” е създаден преди цели 26 години през 1995г., докато “lichess.org” е открит в близката 2010г. Поради по-продължителния си живот “chess.com” е много по-известен и броя на игрите, които се играят в реално време почти постоянно е около 200 000-300 000, докато при “lichess.org” е в пъти по-малко – към 30 000.

Двете платформи споделят следните функционалности:

* запазване на игри на компютъра чрез специално файлово разширение и обратното - зареждането им в сайта
* турнири
* различни настройки за времето, за което се изпълнява ход в игра
* 8 различни вида игра на шах
* игра срещу компютър
* компютърен треньор, работещ под формата на пъзели за решаване и рейтинг на развитието на потребителя
* голям набор от обучителни видеа

В общите функционалности “chess.com” превъзхожда с малка разлика “lichess.org” във частта със обучителните видеа, но за сметка на това “lichess.org” може да излъчва игра на живо за специално поканени потребители, а “chess.com” няма такава опция.

Ако трябва да изберем един от двата, решението зависи изцяло от предпочитанията на конкретния потребител. В безплатната версия на “chess.com” не са включени всички функционалности, като сайта предлага три опции за платен абонамент – златен, платинен и диамантен. Платформата съперник не предлага същото качество на някои от функционалностите, но всичко в нея е напълно безплатно и няма никакви реклами, което я прави много привлекателна опция.

**ВТОРА ГЛАВА**

**ПРОЕКТИРАНЕ НА СТРУКТУРАТА НА WEB ПРИЛОЖЕНИЕ ЗА ИГРА НА ШАХ**

**2.1. Функционални изисквания към WEB приложение за игра на шах**

- Поддръжка на потребители

- Играене на шах в мултиплеър

- История на изиграните игри

- Рейтинг система на играчите

- Поддръжка на турнири

- Чат между играчите по време на играта

Обобщение: приложението трябва да представлява сайт, в който да се регистрират и да влизат в профила си потребители, които да могат да играят на шах един срещу друг. Потребителите трябва да имат комуникация помежду си под формата на чат по време на играта. Нужна е и по-сложна форма на игра, а именно чрез формата на турнир. Трябва да има рейтинг на всеки играч и история на изиграните му игри, която да позволява преглеждането на изиграна игра отново и отново.

**2.2. Съображения за избор на програмни средства и развойната среда**

Технологиите, които избрах за дипломната си работа са следните: Python, Flask, SQLAlchemy, HTML, CSS, JavaScript.

Python е един от най-популярните, развити и функционални програмни езици, а комбиниран с Flask e чудесен за изграждане на WEB приложения.

SQLAlchemy представлява СУБД, която много добре се интегрира в кода за приложението, тъй като разчита на специални структури, които позволяват работата с базата данни без чист SQL скрипт.

Комбинацията на HTML, CSS и JavaScript е абсолютно необходима, когато трябва да се изгради динамичен и функционален сайт към приложението.

Развойната среда, която използвам е съставена от две части: Visual Studio Code и Ubuntu WSL 2.

Използвам Visual Studio Code като редактор за кода, защото като програма редактор е лека, има много функционалности които са лесно достъпни, a интерфейса е богат.

Ubuntu WSL 2, който се използва на Windows 10 представлява алтернатива на вградения в операционната система Linux Ubuntu терминал, като съдържа в себе си същите функционалности и улеснява работата с Flask и Git, както и цялостното управление на компютърната система.

**2.3. Проектиране на структурата на базата от данни**

****

Първата таблица се казва „users“ и съдържа основни данни за потребителите в платформата:

* *id* – идентификационен номер (първичен ключ)
* *username* – име
* *password* – парола (хеширана)

Дали потребителят:

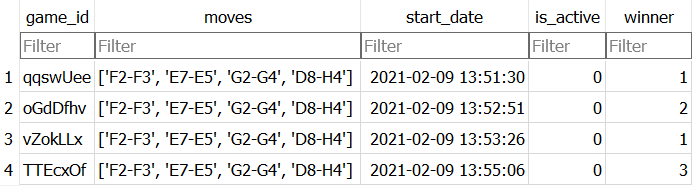
* *is\_logged* – е влязъл в акаунта си
* *is\_waiting* – чака да започне игра
* *is\_playing* – играе в текущия момент

Втората таблица е допълнение на първата и се казва „user\_stats“. В нея се запазват следните данни:

* *user\_id* – идентификационен номер на потребителя (външен ключ)
* *played\_games* – списък с идентификационните имена на всички игри, които потребителя е изиграл
* *win\_rate* – процент на победи

Третата таблица е за игрите (казва се „games“), като тя съдържа:

* *id* – идентификационно име (7 произволни символа – първичен ключ)
* *******w\_player* – идентификационен номер на белия играч (външен ключ)
* *b\_player* – идентификационен номер на черния играч (външен ключ)
* *tournament\_id* – идентификационен номер на турнир, към който принадлежи играта (ако има такъв)

****

Четвъртата таблица е допълнение към „games“ – „game\_details“. В нея се съдържат следните колони:

* *game\_id* – идентификационно име на играта (външен ключ)
* *moves* – списък със всички ходове, направени по време на играта
* *start\_date* – точната дата и час на започване на играта
* *is\_active* – дали играта е активна
* *winner* – идентификационен номер на победителя (външен ключ)

****

Последната таблица се казва „tournaments“ и съдържа данни за създадените турнири. Колоните са следните:

* *id* – идентификационен номер на турнира
* *waiting\_users* – списък с идентификационните номера на чакащите потребители за започване на турнира (щом започне чакащите биват изтрити)
* *quarter\_final* – списък с идентификационните номера на потребителите, започващи игра в четвърт финал
* *semi\_final* – списък с идентификационните номера на потребителите, започващи игра в полуфинал
* *final* – списък с идентификационните номера на потребителите, започващи игра в четвърт финал
* *winner* – идентификационен номер на потребителя, който е спечелил турнира

**ТРЕТА ГЛАВА**

**ПРОГРАМНА РЕАЛИЗАЦИЯ НА WEB ПРИЛОЖЕНИЕ ЗА ИГРА НА ШАХ**

**ЧЕТВЪРТА ГЛАВА**

**РЪКОВОДСТВО ЗА ПОТРЕБИТЕЛЯ**

Това ръководство е ориентирано главно към потребителите на някаква форма на Linux операционна система (Ubuntu-базирана)

Отворете програмата терминал. За да инсталирате нужните програми, пакети и модули използвайте командите по-долу:

Python:

$ sudo apt update

$ sudo apt install software-properties-common

$ sudo add-apt-repository ppa:deadsnakes/ppa

$ sudo apt install python3.9

Pip:

$ pip install pip

Flask:

$ pip install Flask

$ pip install Flask-Login

$ pip install Flask-Session

$ pip install Flask-API

SqlAlchemy

$ pip install SQLAlchemy

$ pip install Flask-SQLAlchemy

Werkzeug

$ pip install Werkzeug

Трябва да създадете две „env“ променливи във файла “.bashrc”.

Първо отваряте файла през терминала, например с командата:

$ nano ~/.bashrc

След това отивате на най-долния ред на файла и създавате две променливи – „CHESS\_DATABASE“ и „PYTHON\_GAME“, като те ще съдържат пътищата до съответните файлови ресурси. Последната папка “TUES” в примерните пътища е папката на Github репозиторито. Пътищата могат да варират в зависимост от ситемата Ви, така че показания по-долу пример е само образец.

export CHESS\_DATABASE='sqlite:///C:\\TUES\\Github\\TUES\\chess.db'

export PYTHON\_GAME='/mnt/c/TUES/Github/TUES/python\_game'

След това запишете файла с командите Ctrl+X, Y, Enter.

За да се сигурни, че променливите са запазени трайно в системата изпълнете следната команда в терминала:

$ source ~/.bashrc

Вече сте готови да стартирате приложението. Влезте в папката, където са изходните ресурси и изпълнете командата:

$ export FLASK\_APP=app.py

Сега можете да стартирате приложението (всеки път) с командата:

$ flask run

Ако искате да терминирате приложението, използвайте Ctrl+C.

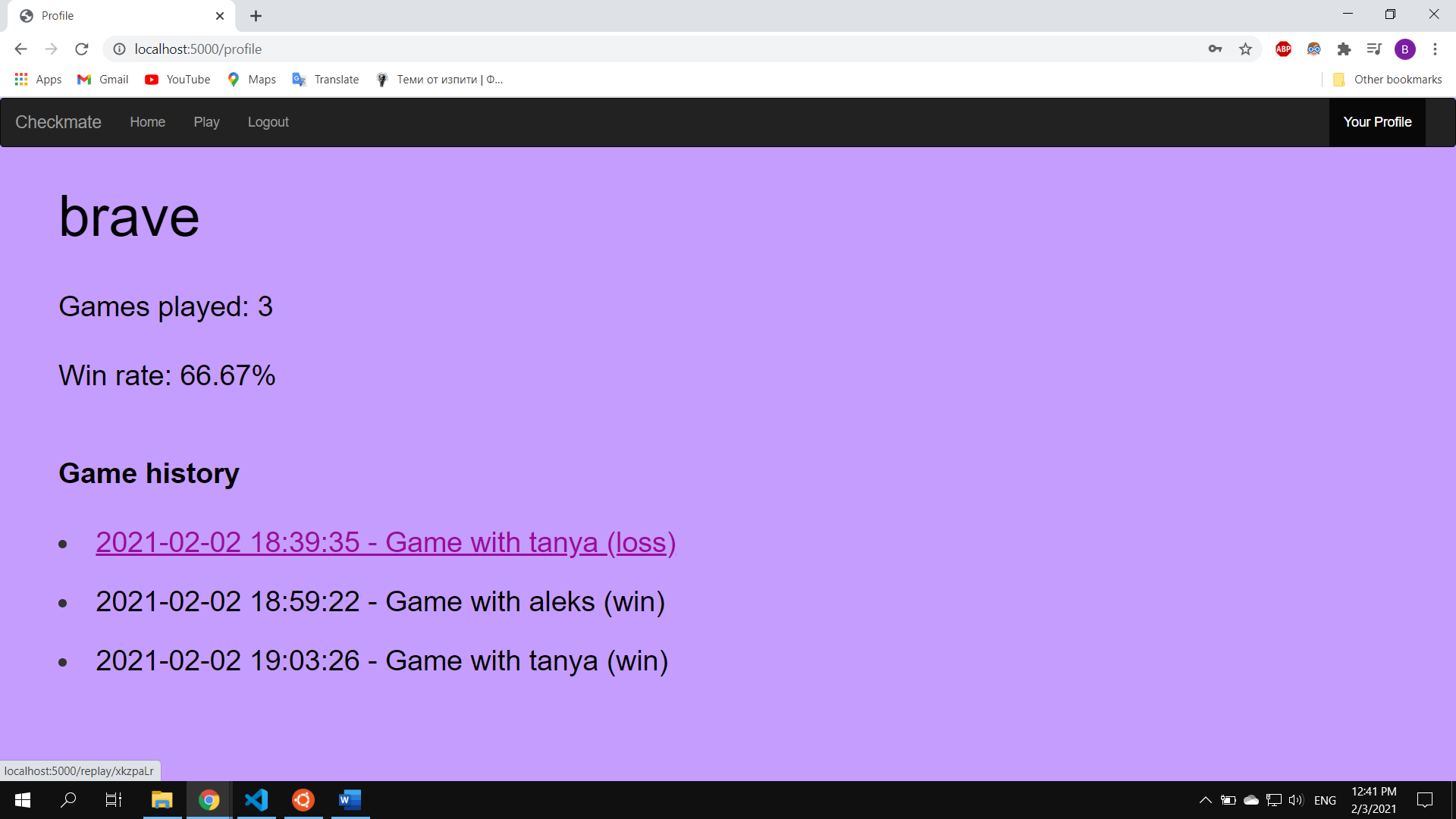


Щом пуснете приложението, заредете адреса на “localhost” в браузъра си и ще се озовете на началната страница.

За да започнете игра първо трябва да влезете в профила си. Кликнете на бутона „Login“ от менюто. За да създадете профил последвайте „Sign up“ линка под полетата.

Щом се регистрирате автоматично ще бъдете пренасочени към “Login” страницата. След въвеждане на данните си ще бъдете пренасочени към началната страница. Ако това не се случи, значи въведените от вас данни не са правилни.

Докато не сте влезли в профила си страницата за игра „Play“ ще ви пренасочва към „Login“. Щом влезете в профила си на менюто за навигация в най-лявата част ще се появи бутон „Your Profile“. Тези функции контролират свободата на потребителите в сайта и го правят по-интуитивен.



Профилната страница се състои от:

* името на потребителя
* брой на изиграните игри
* процент на спечелените игри
* история на игрите

Историята на игрите се състои от списък на изиграните досега игри със следните характеристики:

* датата и часът на започване на играта
* името на потребителя, който е бил опонент
* дали играта е била спечелена, загубена или равен.

Всяка игра от списъка представлява линк към страница, където играта може да бъде проследена ход по ход.



Над дъската пише кой играч с кои фигури е, и бутоните от двете страни преместват фигурите по ходовете, които са изиграни по време на играта.

Когато искате да започнете игра трябва да натиснете бутона „Play“ от горното меню.

На страницата, която се зарежда имате две опции – да играете в нормален мултиплеър или в турнир.

****

Избирайки една от опциите влизате в списъка за чакащите играчи за съответния тип игра. Нормалният мултиплеър започва при двама чакащи, а турнирът – при осем. Отдолу е показан екрана при чакане.

**ЗАКЛЮЧЕНИЕ**

**Обобщение**

Чрез разработването на този проект бе постигната една напълно функционална реализация на WEB приложение за игра на шах, като самата игра представлява ядрото, а сървъра и клиентската част са нужни, за да може играта да бъде използвана по лесен, достъпен и приятен за потребителите начин.

**Бъдещи подобрения**

За да бъде възможна реализацията на проекта на пазарно ниво трябва да се извършат следните промени/допълнителни имплементации:

* да се напишат тестове за сървърните и клиентските функционалности
* да се подобри ефективността на софтуера и максималния размер на данни, които могат да бъдат използвани от системата
* да се имплементира някакъв вид комуникация между потребителите извън конкретна игра, например отбори от играчи и възможност за покана за игра между тях
* да се имплементира изкуствен интелект под формата на бот, който да играе с потребителя на избрано ниво на трудност
* потребителите да се класифицират в лиги, което да улесни намирането на най-добрите играчи в платформата

**ИЗПОЛЗВАНА ЛИТЕРАТУРА**

**За софтуерната част от проекта:**

Flask документация:

<https://flask.palletsprojects.com/en/1.1.x/>

Flask SQLAlchemy документация:

<https://flask.palletsprojects.com/en/1.1.x/patterns/sqlalchemy/>

SQLAlchemy документация:

<https://docs.sqlalchemy.org/en/13/>

jQuery документация:

<https://api.jquery.com/>

Werkzeug документация:

<https://werkzeug.palletsprojects.com/en/1.0.x/>

Не е използвана литература за писмената част на дипломната работа (документацията).

**СЪДЪРЖАНИЕ**

1. Приложени документи
2. Увод
3. Първа глава – Методи и технологии за реализиране на WEB приложения
   1. Основни принципи, технологии и развойни среди за реализиране на WEB приложения
   2. Съществуващи решения и реализации
4. Втора глава – Проектиране на структурата на WEB приложение за игра на шах
   1. Функционални изисквания към WEB приложение за игра на шах
   2. Съображения за избор на програмни средства и развойната среда
   3. Проектиране на структурата на базата от данни
5. Трета глава – Програмна реализация на WEB приложение за игра на шах
6. Четвърта глава – Ръководство на потребителя
7. Заключение
8. Използвана литература