

РУСЕНСКИ УНИВЕРСИТЕТ “АНГЕЛ КЪНЧЕВ”

Факултет: Природни науки и образование
Катедра: Информатика и информационни технологии

ДОПУСКАМ ДО ЗАЩИТА

Ръководител на специализираната катедра:.....
(доц. д-р Румен Русев)

**ОБЯСНИТЕЛНА ЗАПИСКА
НА ДИПЛОМЕН ПРОЕКТ**

Тема:
Discord бот – Studybuddy

Дипломант:
(Д. Дянков)

ОДОБРЯВАМ
Ръководител:

(Б.Иванова)

Русе, 2023 г.

РУСЕНСКИ УНИВЕРСИТЕТ “АНГЕЛ КЪНЧЕВ”

Катедра: Информатика и информационни технологии

Регистрирал

Факултет: ПНО

Инспектор:

(М. Чолакова)

...../....., 2023 г.

УТВЪРЖДАВАМ

Р-л на специализи-

ращата катедра:

(доц. д-р Румен Русев)

ЗАДАНИЕ ЗА ДИПЛОМЕН ПРОЕКТ

На студент: **Дилян Пламенов Дянков**

Фак.№ **196256**

Специалност: **Софтуерно инженерство**

ОКС: **бакалавър**

1. Тема: Discord бот - Studybuddy

2. Изходни данни: JavaScript, SQL, HTML, CSS

3. Съдържание на обяснителната записка:

1. Увод
2. Обзор на съществуващите решения
3. Избор на технология
4. Описание на проекта на приложението
5. Ръководство за работа с приложението
6. Изводи и препоръки

Срок за предаване на дипломния проект: 04.06.2023 г.

Зададено на: 14.12.2022 г.

Дипломант:

(Д. Дянков)

Ръководител:

(Б. Иванова)

Съдържание

I. Увод	5
II. Изложение.....	6
Глава 1. Обзор на съществуващи решения.....	6
1. Образователни интерактивни приложения/платформи	6
1.1. Duolingo.....	6
1.2. DragonBox.....	8
2. Образователни чатботове	9
2.1. Какво представлява чатботът?	9
2.2. Чатбот Andy	9
2.3. Предимства на чатботовете в образованието	10
3. Дискорд чатботове	11
3.1. Какво представлява Дискорд?	11
3.2. Чатботовете на Дискорд.....	12
3.2.1. MEE6 – най-известният Дискорд бот.....	12
3.2.2. Jockie Music – най-известният функциониращ музикален бот	14
3.3. Образователни Дискорд чатботове	15
3.3.1. StudyLion.....	15
3.3.2. TriviaBot	16
3.3.3. Countryvia	17
3.3.4. Archimedes.....	18
3.3.5. Geocord.....	18
4. Сравнителна таблица на сходни съществуващи решения	18
5. Обобщения и изводи	19
6. Цели и задачи	19
Глава 2. Избор на технология.....	20
1. Използвани технологии.....	20
1.1. JavaScript (JS).....	20
1.2. Node.js.....	20
1.3. npm.....	21
1.4. Discord.js и discordjs/voice.....	21
1.5. MySQL, SQL и mysql.js.....	21
1.6. phpMyAdmin	22
1.7. XAMPP	23
1.8. HTML	23
1.9. CSS.....	24
1.10. jQuery	24
1.11. dotenv	24
2. Архитектура на системата	25
3. Логически модел на системата.....	26
3.1. Потребителски случаи	26
3.1.1. Диаграма на случаи на употреба.....	26
3.1.2. Сценарии на потребителски случаи	27
4. Описание на потребителските интерфейси	32

Глава 3. Описание на проекта на приложението	35
1. Описание на базата данни.....	35
1.1. Описание на таблиците в БД	35
1.1.1. Таблица „grade“	35
1.1.2. Таблица „subject“	36
1.1.3. Таблица „lesson“	36
1.1.4. Таблица „lesson_content_type“	37
1.1.5. Таблица „lesson_content“	37
1.1.6. Таблица „question“	38
1.1.7. Таблица „answer“	39
1.1.8. Таблица „question_answer“	39
1.1.9. Таблица „exam“	40
1.1.10. Таблица „exam_question_answer“	40
1.2. Релационен модел на БД	42
2. Реализация на проекта	43
2.1. Реализация на приложението StudyBuddy.....	43
2.1.1. Структура.....	43
2.1.2. Програмна реализация на основата (index.js)	44
2.1.3. Програмна реализация на връзката с БД и изпращане на SQL заявки	46
2.1.4. Програмна реализация на командите	48
2.1.5. Програмна реализация на валидация на разрешения	50
2.2. Реализация на уебсайт с наръчник за употреба	51
2.2.1. Структура.....	51
2.2.2. Програмна реализация на уебсайта.....	51
3. Тестване на приложението StudyBuddy.....	52
3.1. Тестване на производителност (performance testing)	52
Глава 4. Ръководство за работа с приложението	56
1. Инсталация и стартиране	56
1.1. Инсталация	56
1.2. Стартиране	56
2. Ръководство за използване	56
2.1. Работа с команда /материали	56
2.2. Работа с команда /уроци	57
2.3. Работа с команда /тестове	58
Глава 5. Изводи и препоръки	61
III. Заключение	61
IV. Литературни източници (конвенция APA)	62

I. Увод

Образованието е основен аспект на човешкото развитие, оформящ знанията, уменията и перспективите на хората. С навлизането в дигиталната ера, технологията продължава да революционизира начина, по който се предоставя образованието, правейки го по-достъпно, интерактивно и ангажиращо. Една от нововъзникващите тенденции в сферата на образованието е използването на чатботове – компютърни програми, предназначени да автоматизират задачи и да взаимодействат с потребителите-човеци. Дискорд бот StudyBuddy е иновативен, образователен инструмент, специално създаден да подобри и улесни учебния процес на учениците в пети, шести и седми клас.

Прогимназиалните години отбелязват критична фаза в академичния път на учениците, характеризираща се с повишена академична строгост и развитие на важни умения и знания. Приложението има за цел да подтикне учениците към самообучение, да насърчи ангажираността ѝм и да осигури цялостна подкрепа по интерактивен и динамичен начин.

Дискорд – популярна комуникационна платформа сред подрастващите – служи като идеална платформа за вместване на предложеното решение. Възползвайки се от възможностите ѝ, ботът създава виртуална учебна среда, в която учениците могат да получат достъп до образователни ресурси и да извършват самооценка на придобити знания.

Приложението разполага с четири функции: общ достъп до всички учебни материали според възложени критерии; визуализация на цялостно структурирани уроци, състоящи се от всички материали, свързани с конкретния урок (текст, изображения, видеоклипове и аудиоклипове); възможност за инициране на тестове с цел проверка нивото на знания на ученика; получаване на хипервръзка към подробно описано ръководство за употреба на програмата.

Целта на дипломния проект е разработване на образователен Дискорд бот, който да подпомогне учениците от 5 до 7 клас чрез по-лесно усвояване и придобиване на знания в областите география и икономика, история и цивилизации и информационни технологии. Задачите, поставени пред проекта са: разработване на приложение с интерактивен интерфейс, подходящ за деца; възможност за възпроизвеждане на мултимедия; изграждане на база от данни, съдържаща учебните материали; възможност за извършване на самооценяване; разработване на уебсайт с ръчник за употреба, представен на подходящ език за учениците.

II. Изложение

Глава 1. Обзор на съществуващи решения

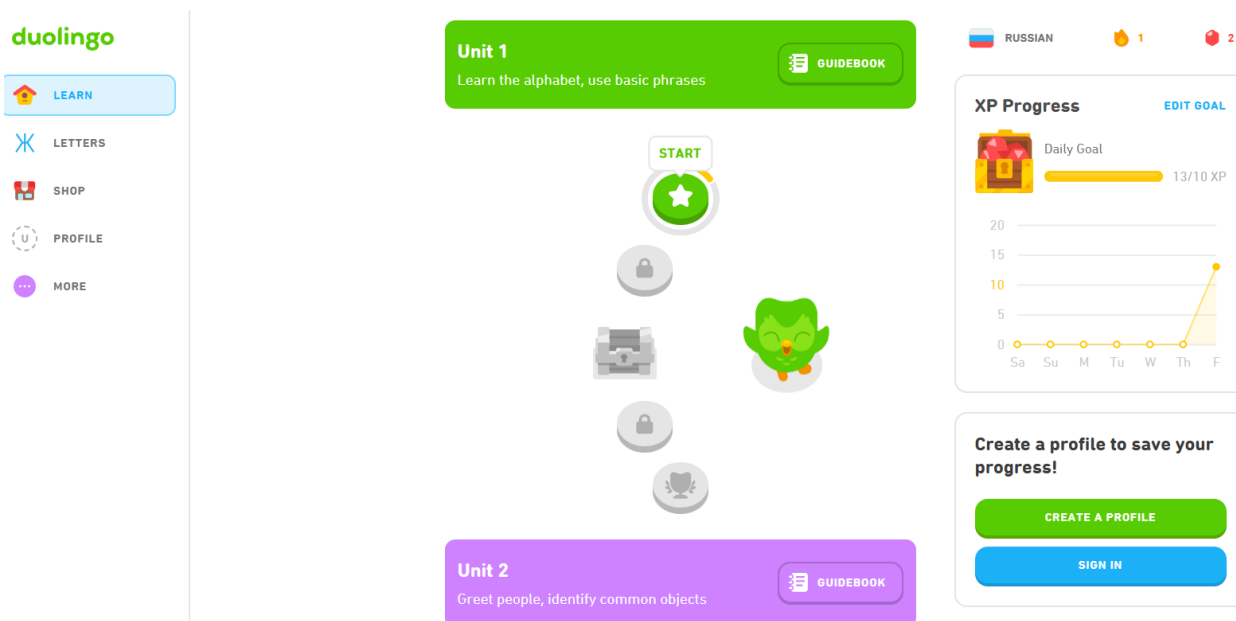
1. Образователни интерактивни приложения/платформи

1.1. Duolingo



Duolingo е платформа за учене на езици, която предлага курсове на над 40 езика. Платформата може да се достъпи чрез мобилно приложение или през уеб браузър. Duolingo използва игрови подход, за да направи изучаването на нов език забавно и ангажиращо. Някои от функциите му включват персонализирано обучение, което се адаптира към нуждите на всеки потребител, кратки и фокусирани уроци, които се вписват в ежедневието, упражнения за слушане, говорене, четене и писане, мултимедийни ресурси за „потапящо“ учебно изживяване, възможност за проследяване на напредъка и социални функции за свързване с приятели. Платформата също така предлага изпити за езиково сертифициране, признати в цял свят. Duolingo предоставя достъпен и приятен начин за изучаване на нов език за всички учащи, независимо от тяхното ниво или възраст.[1]

Разработчиците на Duolingo предоставят достъп до платформата още със зареждането на интернет страницата. Регистрацията е безплатна и незадължителна, но е препоръчителна, за да бъде възможно запазването на прогреса по време на обучение. Потребителите на Duolingo разполагат с табло за управление, чрез което получават достъп до уроци, упражнения, тестове, статистики и настройки. (Фиг.1)

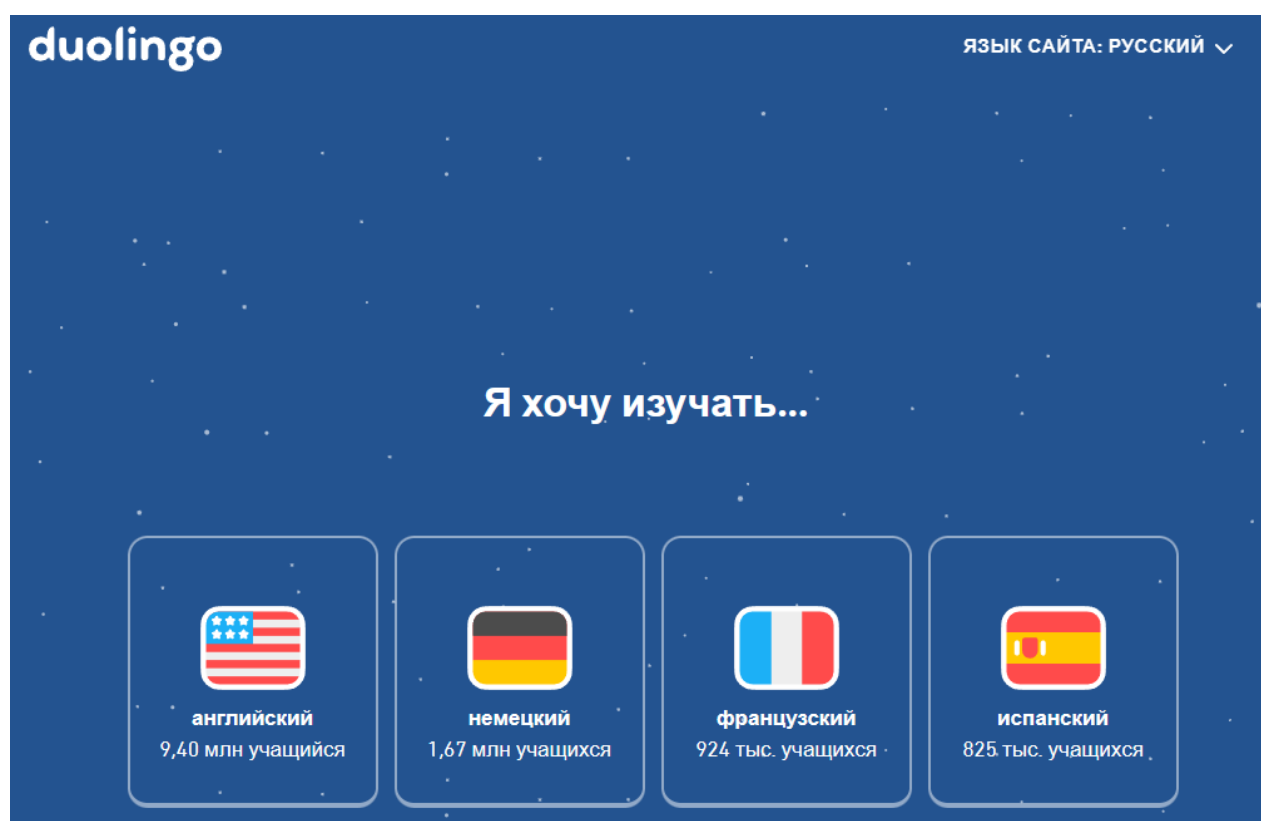


Фиг. 1: Табло за управление (dashboard) на Duolingo

Предимствата на платформата се състоят в това, че тя е безплатна, леснодостъпна и предлага учене на множество езици едновременно. Потребителите имат

възможността да използват платформата навсякъде и по всяко време, стига да разполагат със смартфон и интернет връзка. Голяма част от успехът на Duolingo се дължи на факта, че ученето се осъществява напълно безплатно. Много хора по света имат желание да изучават чужди езици, но не разполагат с финансовите средства, които обикновено се изискват при традиционните методи на обучение. За разлика от други приложения за изучаване на езици, Duolingo не ограничава потребителите си по отношение на това колко езика могат да учат едновременно.

Въпреки предимствата си, платформата има и множество недостатъци. Тя функционира на английски език и предлага изучаване на 40 езика, но за да може потребителят да се възползва от възможностите на платформата, той трябва да познава английски език. Ако например потребителят познава единствено руски език, вариантите за избор на език за учене са само 4. **(Фиг. 2)** Като друг недостатък може да бъде определено нивото на обучение в платформата – обучението в Duolingo е ориентирано към начинаещи, което означава, че е невъзможно потребителите да надграждат придобити знания в областта на някой език.[2]



Фиг. 2: Възможни езици за изучаване за рускоговорящи потребители

1.2. DragonBox

DragonBox е серия от образователни детски игри, предназначени да преподават математически концепции по забавен и интерактивен начин. Игрите са налични за мобилните операционни системи iOS и Android. Първата игра от поредицата – DragonBox Algebra 5+ – е създадена през 2012 г. Оттогава серията се е разширила, като днес игрите, чрез които децата могат да учат числа, алгебра и геометрия, вече са общо 5. Всяка игра е проектирана да бъде достъпна и увлекателна за децата, без да са необходими предварителни познания по математика. Вместо текст, за обучение на учениците игрите DragonBox използват картини и символи (**Фиг. 3**). Това позволява на децата, които все още не са се научили да четат, да могат да играят и същевременно да се учат от игрите. Серията е носителка на множество награди и през годините е била хвалена за иновативния си подход към преподаването на математика.[3]



Фиг. 3: Демонстрация на DragonBox Algebra 5+

Серията DragonBox включва следните игри:

- **Algebra 5+** – Това е първата игра от поредицата. Предназначена е да преподава алгебра на деца на възраст минимум пет години. Играта използва визуален подход, като играчите решават пъзели, като местят обекти на екрана, за да изолират променлива;
- **Numbers** – Тази игра учи децата на смисъла на числата и основните аритметични операции, като използва серия от интерактивни пъзели и игри. Играчите се учат да броят, събират, изваждат и умножават по забавен и увлекателен начин;

- **Big Numbers** – Играта е предназначена за малко по-големи деца, които вече са усвоили основната аритметика. Играчите се научават да използват по-големи числа и трябва да решават все по-сложни пъзели, използвайки различни математически операции;
- **Elements** – Приложението учи децата на геометрични концепции, използвайки подобен визуален подход на играта Algebra 5+. Играчите решават пъзели, като местят фигурки на екрана, като по този начин научават за свойствата на различните форми;
- **Algebra 12+** – Тази игра обучава деца на възраст минимум дванадесет години, като ги учи на алгебра с по-висока трудност. Играта е разделена на десет части, като всяка част се фокусира върху различна алгебрична концепция, като променливи, уравнения и неравенства. Играта започва с основни концепции, като постепенно повишава нивото на трудност с напредъка на играча.[4]

2. Образователни чатботове

2.1.Какво представлява чатботът?

Чатбот е компютърна програма, предназначена да симулира разговор между хора чрез текстови или гласови комуникационни канали. Те използват изкуствен интелект и техники за обработка на естествен език, за да разберат потребителските заявки и да предоставят подходящи отговори. Чатботовете имат приложения в различни области – могат да бъдат използвани за обслужване на клиенти, за маркетинг, за личен асистент и т.н. Ботовете подпомагат автоматизирането на задачи или предоставянето на информация.

Чатботовете едва наскоро започнаха да навлизат в областта на образованието, поради което съществуват минимален брой решения, разработени с конкретната цел да образоват.

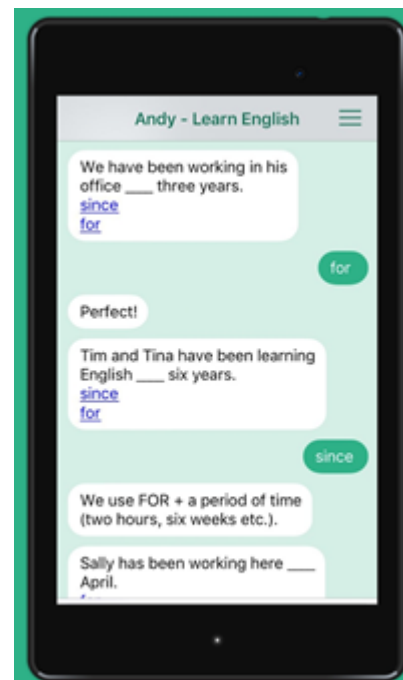
2.2. Чатбот Andy

Ботът Andy е създаден специално за изучаване и упражняване на английски език. Потребителите могат да провеждат дискусии или да играят езикови игри. Ботът разполага с опции за изучаване на английската граматика, както и за разширяване на речниковия запас на потребителите. Комуникацията между потребителя и приложението се осъществява единствено чрез текстови съобщения; не се поддържа гласова комуникация.

Учещите английски език могат да намерят обяснения за различни граматически правила, като в края на всеки урок Andy проверява знанията им чрез изпълнение на тест. Ако потребителят допусне грешка, чатботът му съобщава за това и му обяснява



какво точно е сгрешил (**Фиг. 4**). Допуснатите грешки биват упоменати дори по време на обикновени разговори (ботът постоянно следи за грешки в съобщенията на потребителите). Потребителите имат възможността да научат нови думи във всеки един момент, в който пожелаят. За целта, те единствено трябва да уведомят Andy, след което той ще съобщи дефинициите на думите. В допълнение, потребителят получава и примери за думите с контекст от реалния живот.[5]



Фиг. 4: Примерен разговор между потребител и Andy

2.3. Предимства на чатботовете в образованието

➤ Висока мотивация

Много приложения за чат използват игровизация[6] (прилагане на игрови елементи и механики в неигрови ситуации). Например, потребителите печелят награди, значки или някакъв вид постижения, след като използват приложението няколко последователни дни или след като постигнат някаква цел. По този начин потребителят се чувства награден и удовлетворен от себе си. Тези функции правят чатботовете забавни за използване, а когато човек се забавлява по време на учене, той е по-склонен да запомня.

➤ Подходящи за хора със социална тревожност

Чатботовете са винаги дружелюбни и не съдят потребителите си. Те не въртят очи и не поклащат глава, когато човек допусне грешка. Също така, по време на разговор между бот и потребител липсват третите лица, които слушат разговора и най-вероятно биха съдили потребителя, ако допусне грешка. Пример за такива лица са именно съучениците на един ученик по време на учебно занятие в училище. По време на комуникация с приложението едно дете не би се чувствало притеснено да сбърка. Това прави чатботовете изключително полезни и подходящи за обучение на подрастващи.

➤ Персонализирано изживяване

В класната стая учениците обикновено трябва да изчакат своя ред, за да говорят. Едно дете рядко получава възможност да се упражнява, като говори очи в очи с учител и получи обратна връзка за произношението и граматиката си. Чатботове разрешават този проблем, като предоставят персонализирано и индивидуализирано обучение.

➤ Винаги на разположение

Поради естеството на приложенията за чат, потребителите имат възможността по всяко време и навсякъде да стартират приложението на своите устройства и да се впуснат в разговори с техният виртуален приятел и учител.

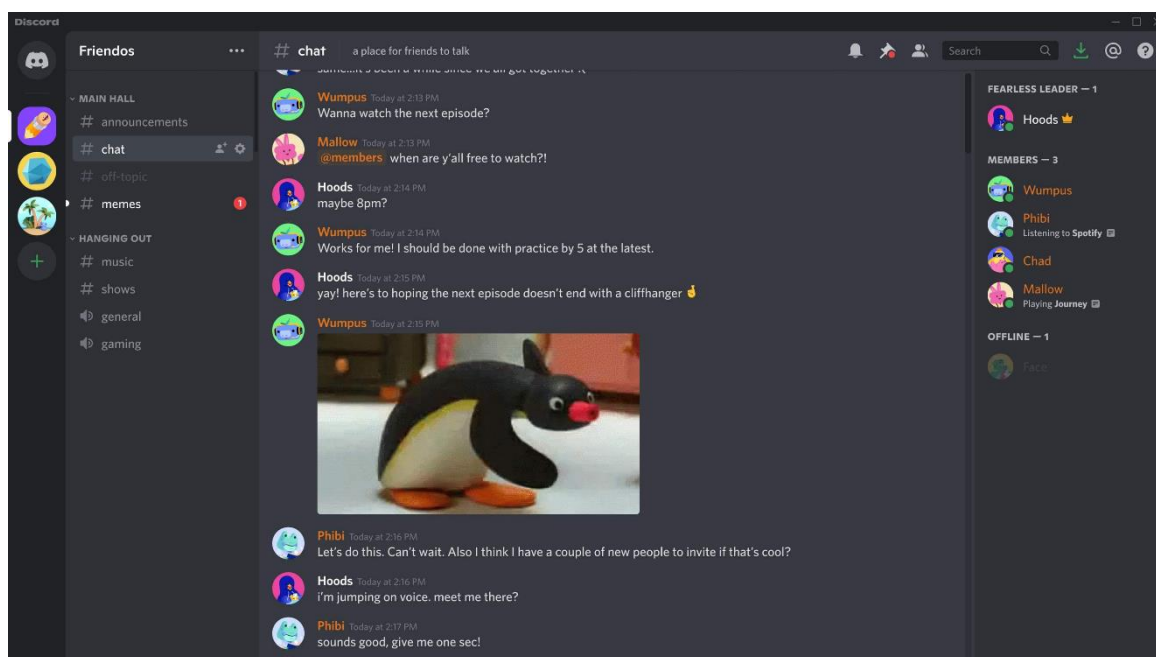
3. Дискорд чатботове

3.1. Какво представлява Дискорд?

Дискорд е безплатна платформа за комуникация, която позволява на потребителите да обменят текстови съобщения и да провеждат гласови и видео разговори с други потребители в реално време. Първоначално Дискорд бе предназначен за геймъри, но впоследствие бе разширен, като днес се използва от множество различни видове общности (преподаватели, художници, бизнесмени, любители и т.н.).



Потребителите могат да създават или да се присъединяват към сървъри, които по същество представляват онлайн общности, съсредоточени около определена тема или интерес (**Фиг. 5**). Потребителите имат способността по всяко време да превключват между сървърите, в които членуват. Всеки сървър се състои от един или повече текстови и/или гласови канали, посветени на конкретни теми или дискусии, които потребителите могат да създават. Каналите могат да бъдат използвани за всичко - от всеобщ чат до стратегически дискусии за конкретна игра.



Фиг. 5: Примерен изглед тип „Сървър“ в Дискорд

Освен сървъри и канали, Дискорд поддържа и директен обмен на съобщения между потребители. Тази функция може да се използва за лични съобщения или за групов чат с избрано множество от хора.

Една от ключовите характеристики на платформата е нейната възможност за провеждане на гласови и видео разговори. Потребителите могат да инициират тези разговори с други хора в същия сървър или в раздела „Директни съобщения“. В допълнение, Дискорд предоставя възможност на своите потребители да споделят

екрана си, с цел по-голямо удобство по време на различните видове дейности (напр. сътрудничество по време на решаване на даден проблем).

Приложението също така разполага и с набор от други функции, като възможност за споделяне на файлове и връзки, създаване на анкети, използване на емотикони, добавяне на „реакции“ към съобщение и интегриране с други приложения и услуги.

3.2. Чатботовете на Дискорд

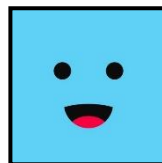
Дискорд чатботовете се превръщат в основен инструмент за управление и подобряване на онлайн общностите. Те предлагат редица функционалности, които могат да бъдат от голяма полза при автоматизиране на рутинни задачи, насърчаване на ангажираността у потребителя и рационализиране на комуникацията. Поради това те стават все по-популярни сред ползвателите на Дискорд от всякакъв вид - от геймъри до бизнес професионалисти.

Чатботовете в Дискорд са предназначени да предоставят на потребителите денонощна поддръжка, като гарантират, че нито едно запитване няма да остане без отговор и нито една задача няма да остане неизпълнена. Автоматизирайки рутинни задачи като модерирание и планиране на събития, те позволяват на сървърните модератори (които са човешки същества) да се съсредоточат върху по-важни задачи като насърчаване на ангажираността и изграждане на общност. В допълнение към практическите си ползи, ботовете могат да осигурят и по-персонализирано преживяване за потребителите, като отговарят на запитвания в реално време и предлагат персонализирани препоръки въз основа на поведението на потребителите.

Едно от най-значимите предимства на чатботовете е способността им да създават усещане за сплотеност в онлайн средата, в която се намират. Предоставяйки интерактивни функции като игри и викторини, ботовете подканват потребителите да взаимодействат помежду си. Ботовете умеят да задържат потребителите ангажирани чрез предоставяне на редовни актуализации за събития и нови функции, създавайки чувство на очакване и вълнение. Тъй като все повече онлайн общности в Дискорд започват да употребяват чатботове, стана ясно, че те имат потенциала коренно да променят начина, по който потребителите взаимодействат онлайн.

3.2.1. МЕЕ6 – най-известният Дискорд бот

Използван от близо 20 милиона сървъри, МЕЕ6 е най-популярният чатбот сред онлайн пространството на Дискорд. Той предоставя различни функции, чрез които помага на потребителите на платформата да управляват своите сървъри.

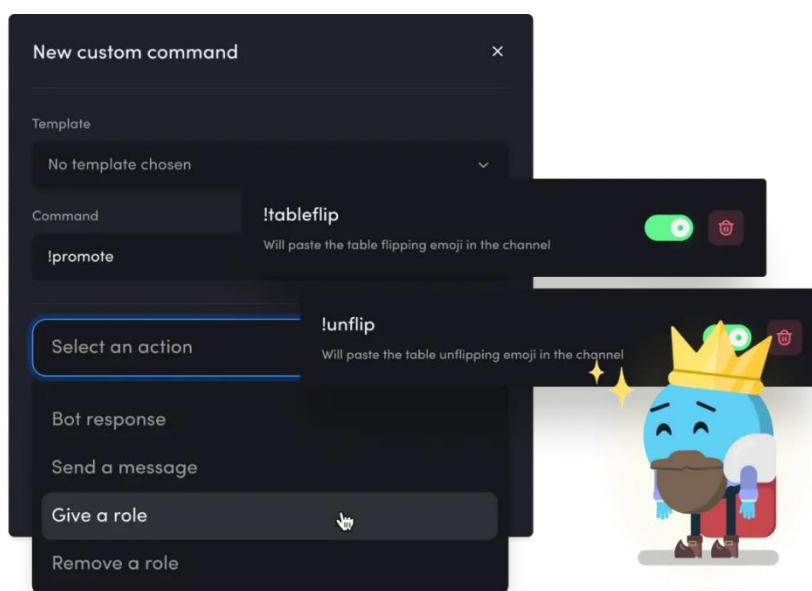


Една от ключовите характеристики на бота е неговата възможност за автоматизирано модерирание. Ботът може да открива и управлява спам съобщения, вулгарен език и хиперлинкове, което позволява на създателите на сървъри да поддържат сървърите си чисти и безопасни за всички членове. Тази функция може да бъде персонализирана по преценка на собствениците на сървъра.

Друга важна характеристика на МЕЕ6 е наличието на система за „вдигане на ниво“ (т.нар. level-up system). Изпълнявайки различни дейности в сървъра (чат, участие в гласови канали, ...), потребителите получават т.нар. точки опит (*experience points; XP*). Печелейки точки опит, потребителите „вдигат нивото си“, като по този начин отключват награди или специални роли. Тази функция помага за насърчаване на активност в рамките на сървъра.

Този чатбот предлага и възможност за създаване на персонализирани команди (**Фиг. 6**). Членовете на сървъра могат да настроят команди, които МЕЕ6 ще изпълни при задействане. Това може да бъде полезно при необходимост от извличане на информация или предоставяне на автоматизирани отговори.

В допълнение към гореспоменатите основни функции, ботът разполага и с няколко други полезни инструменти. Например, той може да се използва за управление на потребителски роли, посрещане на нови потребители и проследяване на потребителската активност. Налични са и различни настройки за персонализиране, с цел адаптиране на поведението на МЕЕ6 към нуждите на всеки отделен сървър.[7]



Фиг. 6: Процес по добавяне на персонализирана команда

3.2.2. Jockie Music – най-известният функциониращ музикален бот

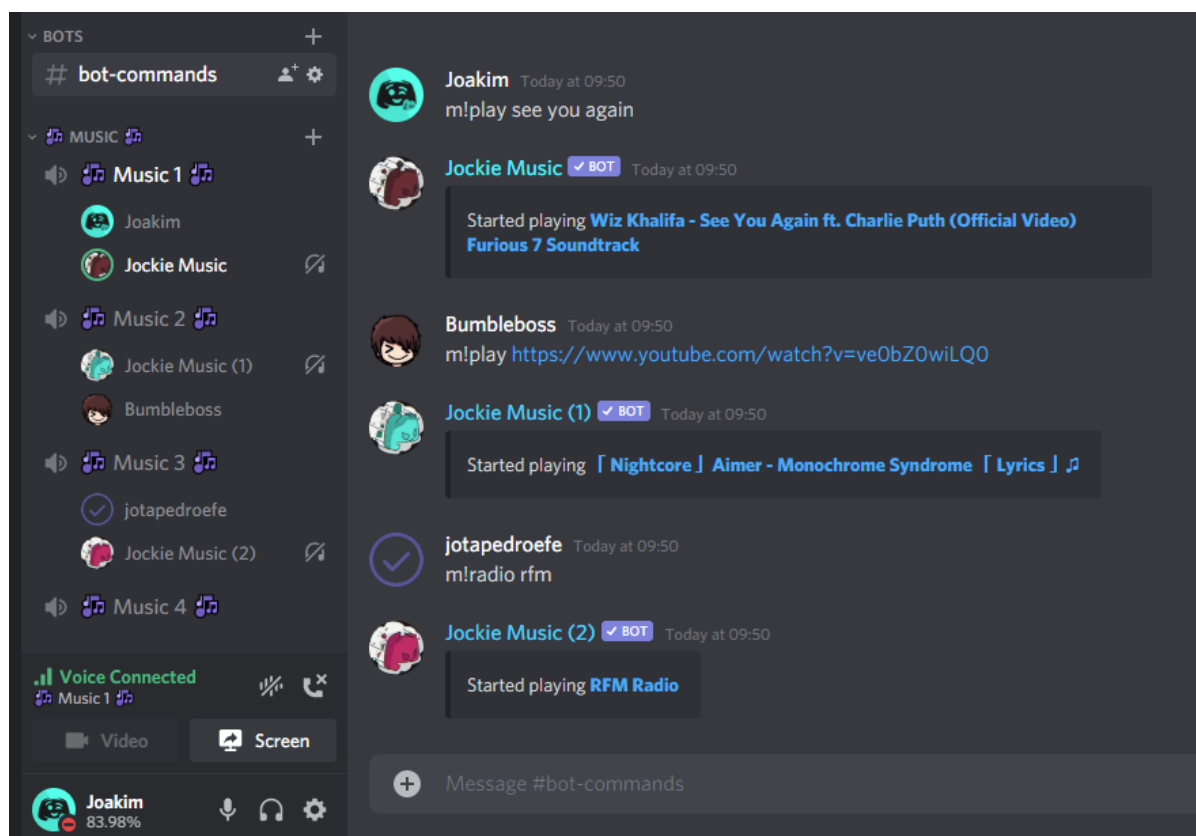


С над 2,4 млн. добавяния, Jockie Music е най-разпространеният бот на Дискорд, изцяло посветен на предоставяне на различни функции, свързани с възпроизвеждането на музика.

Една от основните му характеристики е способността му да възпроизвежда музика от различни източници, включително Spotify, Apple Music, Soundcloud, Vimeo и други (**Фиг. 7**). Тази функция е особено полезна за сървъри, посветени на музиката, тъй като позволява на членовете на сървъра лесно и бързо да споделят и слушат любимите си песни.

Освен възпроизвеждане на музика, Jockie предлага редица други полезни функции. Например, ботът може да се използва за създаване на плейлисти или за поставяне на песни в опашка с цел по-късно възпроизвеждане. Също така, потребителите могат да променят настройките за възпроизвеждане на музика на бота, като сила на звука или ред на възпроизвеждане на песните в опашката.

Подобно на MEE6, този чатбот предлага редица опции за персонализиране, като възможност за настройка на персонализирани команди или псевдоними. По този начин потребителите получават възможността да създават преки пътища към често използваните от тях функции, което им осигурява значително улеснение в ежедневието.



Фиг. 7: Процес по възпроизвеждане на музика чрез Jockie Music

По отношение на модерирането, Jockie Music разполага с инструменти, които помагат на собствениците на сървъри да предотвратят злоупотреба при възпроизвеждането на музика. Тези инструменти могат да се използват за управление на потребителски разрешения, налагане на ограничения на опашката и автоматично прескачане на песни, които съдържат вулгарен език или друго неподходящо съдържание.[8]

3.3. Образователни Дискорд чатботове

3.3.1. StudyLion

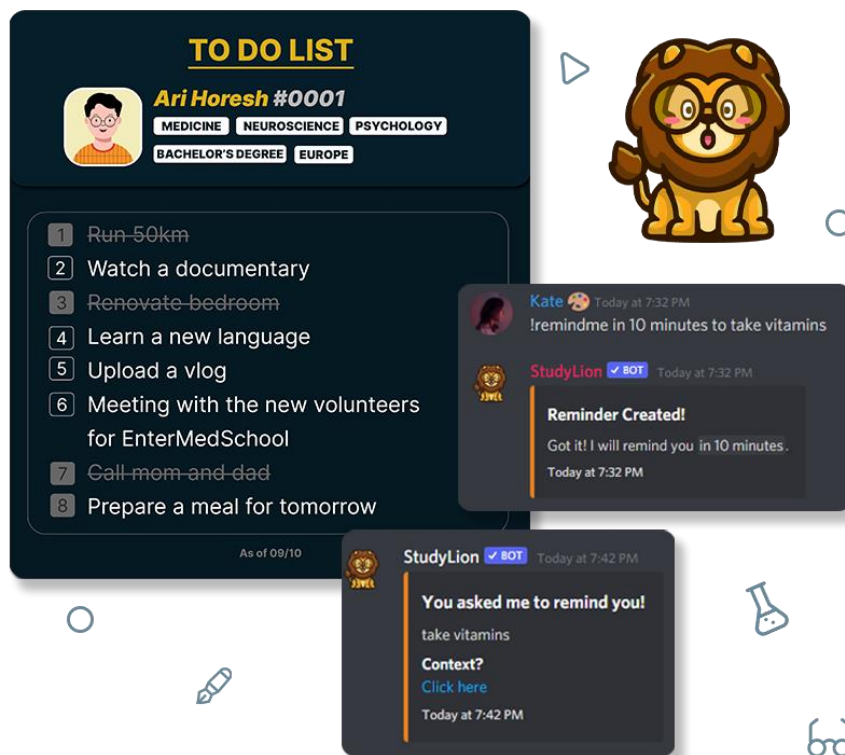
StudyLion е бот, предназначен да подобри учебното преживяване на учениците в Дискорд. Той предлага набор от функции, които позволяват на учениците да бъдат добре организирани, да управляват времето си и да имат достъп до учебни ресурси.



Част от характеристиките на чатбота е способността му да създава и управлява виртуални учебни зали. Тези зали могат да бъдат създавани въз основа на предпочитания от потребителя предмет, ниво на обучение и наличност на време. По време на занятието, членовете на виртуалната зала разполагат с текстови канал, за да могат да общуват и да си сътрудничат в процеса на изпълняване на задачи и проекти.

Друга ключова характеристика на StudyLion е неговият инструмент за планиране. Чрез него, учениците имат възможността да планират напомнания за предстоящи задания, тестове и други важни събития (**Фиг. 8**). Това позволява на учениците да останат организирани и да избегнат пропускането на важни крайни срокове.

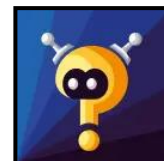
Ботът разполага с набор от функции за производителност, като списък със задачи (To-Do lists) и инструмент за управление на времето. Списъкът със задачи позволява на учениците да следят задачите си и да ги приоритизират според важността им, а функцията за управление на времето им помага да разпределят времето си ефективно. Приложението може да предостави персонализирани препоръки въз основа на нивото на обучение и интересите на ученика.[9]



Фиг. 8: Примерен списък със задачи и процес по добавяне на напомняне

3.3.2. TriviaBot

TriviaBot е образователен бот, който образова своите потребители чрез интерактивни игри. Тези игри съдържат любопитни факти и въпроси, които позволяват на потребителите да тестват знанията си в различни направления (Фиг. 9). Ботът предлага набор от функции, които правят играта увлекателна и предизвикателна за играчите.

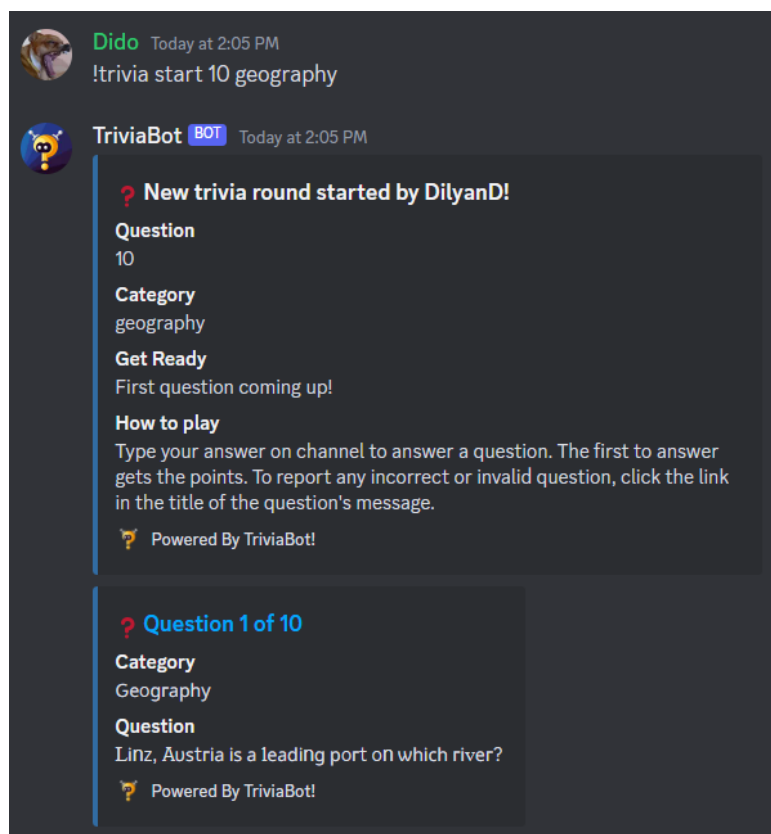


Една от особеностите на TriviaBot е неговата огромна база данни с любопитни въпроси. Базата данни е изпълнена с въпроси от широк спектър от теми, включително история, география, математика, физика, биология и много други. Тези въпроси се актуализират редовно, за да се гарантира, че потребителите винаги са предизвикани и ангажирани.

Чатботът предоставя на своите потребители достъп до персонализирани настройки. Потребителите могат да избират от кои категории да бъдат въпросите, които да получават, нивото на трудност и броя въпроси за рунд. По този начин, потребителите имат способността да приспособят играта към своите предпочитания и ниво на знания.

TriviaBot предлага и мултиплейър режим, при който потребителите могат да се състезават един срещу друг в реално време. Ботът следи резултата на всеки играч и обявява победителя в края на играта. Това добавя състезателен елемент към играта и я прави по-ангажираща за играчите.

В допълнение, приложението предоставя подсказки и обяснения за всеки въпрос, което помага на играчите да се учат и да разширят знанията си. Съществува и класация на най-добрите играчи, което може да мотивира потребителите да продължат да се учат, за да подобрят своите резултати.[10]



Фиг. 9: Стартиране на игра, състояща се от 10 въпроса от категория география

3.3.3. Countryvia

Подобно на TriviaBot, Countryvia е образователен бот, който обучава своите потребители чрез игри с въпроси, фокусирани върху географската научна област. За разлика от предходните две приложения, този бот е семпъл и беден на функции.



Ботът разполага само с една функция, която е именно задаването на въпроси. Въпросите могат да бъдат от три категории: знамена, столици и карти. В процеса на стартиране на игра, потребителят въвежда желан брой въпроси, както и една от трите категории, която е избрал.

Въпреки простотата на приложението, то също може да се окаже полезно за научаване на нови факти за различни държави по света, тъй като използва доказано ефективния метод игровизация.[11]

3.3.4. Archimedes

Archimedes е бот, чрез който потребителите са могли да упражняват математическите си умения. Математическите операции варират от обикновено събиране до решаване на квадратни уравнения. Ботът е разполагал с функция за проследяване на правилните и грешните отговори, както и с функция за показване на изчерпателни обяснения на потребителите, включително междинни стъпки. За съжаление, днес Archimedes не функционира.[12]

3.3.5. Geocord

Сходен на Countryvia, Geocord е образователен бот, свързан с учене на географски факти чрез интерактивни игри. Тези игри са представлявали въпроси, на които потребителите е трябвало да отговорят. Всяка игра е можело да бъде играна в един от три режима: „карти“, „знамена“ и „по-високо или по-ниско“. При избор на режим на игра „по-високо или по-ниско“, ботът е визуализирал две държави, след което играчите е трябвало да познаят коя от двете има по-голям брой население. За нещастие, към този момент чатботът не работи.[13]

4. Сравнителна таблица на сходни съществуващи решения

Въз основа на разгледаните дотук съществуващи решения, техните прилики и разлики могат да бъдат представени в табличен вид (**Таблица 1**).

Ботове	StudyLion	TriviaBot	Countryvia	Archimedes	Geocord
Характеристики					
Функциониращ	ДА	ДА	ДА	НЕ	НЕ
Безплатна версия	ДА	ДА	ДА	ДА	ДА
Предлага повече функции срещу заплащане	НЕ	ДА	НЕ	НЕ	НЕ
Инструмент за планиране	ДА	НЕ	НЕ	НЕ	НЕ
Инструмент за управл. на времето	ДА	НЕ	НЕ	НЕ	НЕ
Използва игровизация	НЕ	ДА	ДА	ДА	ДА
Списък със задачи	ДА	НЕ	НЕ	НЕ	НЕ
Възможност за персонализиране на настройки	ДА	ДА	НЕ	НЕ	НЕ

Таблица 1: Сравнение на характеристиките на споменатите Дискорд ботове

5. Обобщения и изводи

Предвид разгледаната в точка 4 сравнителна таблица на съществуващи решения, можем да направим следните заключения:

- ❖ Всички приложения разполагат с безплатна версия;
- ❖ Не всички приложения са функциониращи;
- ❖ Не всички приложения предлагат възможност за персонализиране според нуждите на потребителя;
- ❖ Повечето приложения употребяват игровизация.

Приложението StudyBuddy е едно иновативно решение, като понастоящем **не съществува негов еквивалент**.

6. Цели и задачи

Целта на настоящия проект е разработване на образователен Дискорд бот, който да подпомогне учениците от 5 до 7 клас чрез по-лесно усвояване и придобиване на знания в областите география и икономика, история и цивилизации и информационни технологии.

За реализирането на целта трябва да бъдат изпълнени следните задачи:

- ❖ Разработване на приложение с привлекателен и интерактивен интерфейс, подходящ за деца;
- ❖ Възможност за възпроизвеждане и визуализация на мултимедия – текст, изображения, видеоклипове, аудиоклипове;
- ❖ Изграждане на база от данни, съдържаща учебни материали по география и икономика, история и цивилизации и информационни технологии;
- ❖ Възможност за извършване на самооценяване върху направление по избор;
- ❖ Разработване на уебсайт, съдържащ наръчник за употреба, представен на достъпен и разбираем език за подрастващите.

Глава 2. Избор на технология

1. Използвани технологии

1.1. JavaScript (JS)

JavaScript е динамичен и интерпретируем език за програмиране от високо ниво, който се използва за създаване на интерактивни уеб страници и приложения. Другото име на езика е ECMAScript, като към момента последната версия е ECMAScript 6. Създаден е от Брендън Айх през 1995 г. и оттогава се е превърнал в един от най-широко използваните езици за програмиране в света.

JS е клиентски скриптов език, което означава, че работи на компютъра на клиента, а не на сървъра. Това му позволява да се използва за създаване на динамични и интерактивни уеб страници, които могат да реагират при подадени входни данни от потребителя и да се актуализират сами, без да изисква опресняване на страницата.

Езикът притежава способността да манипулира документния обектен модел (*DOM – Document Object Model*). Поради тази причина програмистите могат да използват JavaScript за да добавят, премахват или модифицират елементи на уеб страница, да променят стилове и атрибути и да отговарят на потребителски събития като кликания с мишката и натискания на клавиши.[14]

1.2. Node.js

Node.js е многоплатформена среда за изпълнение на сървърни и мрежови приложения с отворен код, изградена върху JavaScript двигателя на Chrome – V8. Създадена е от Раян Дал през 2009 г. и оттогава се превърна в една от най-популярните технологии за изграждане на мащабируеми и високопроизводителни уеб приложения.

Node.js позволява на разработчиците да използват JS от страната на сървъра, което означава, че могат да използват един и същ език за програмиране както за front-end, така и за back-end разработки. Това позволява на програмистите да създават цялостни уеб приложения, използвайки един език, което спестява време и намалява сложността.

Друга характеристика на Node.js е неговият неблокиращ I/O модел, благодарение на който се обработват голям брой едновременни връзки, без да блокира цикъла на събитията. Това прави тази среда идеална за изграждане на уеб приложения, които изискват висока производителност и ниска латентност.[15]

1.3.npm

Npm (Node Package Manager) е мениджърът на пакети по подразбиране на Node.js. Той се използва от милиони разработчици за намиране, споделяне и повторно използване на пакети с код в техните проекти. Npm е най-големият софтуерен регистър в света. Той бе въведен за първи път през 2010 г. и оттогава се превърна в неразделна част от екосистемата на Node.js.

Пакетите на npm обикновено се публикуват от отделни разработчици, проекти с отворен код и компании и могат да варират от малки библиотеки с помощни програми до широкомащабни приложения и frameworks. Пакетите могат да включват всичко - от библиотеки с кодове и инструменти до уеб компоненти и цялостни уеб приложения.

Този мениджър на пакети е ключов инструмент по време на работа с Node.js и играе основна роля в растежа и успеха на екосистемата на Node.js.[16]

1.4.Discord.js и discordjs/voice

Discord.js е мощен модул на Node.js за улеснено взаимодействие с приложно-програмният интерфейс (API) на Дискорд. Използвайки Discord.js, разработчиците имат възможността удобно и лесно да създават ботове и други приложения, които да взаимодействат с потребителите. Модулът бе въведен през 2015 г., като с времето се превърна в най-широко използвания инструмент за създаване на Дискорд ботове.

Discord.js предоставя богат набор от функции за взаимодействие с API-то на Дискорд, като изпращане на съобщения, управление на канали и сървъри и взаимодействие с потребители. Той също така осигурява програмируеми действия при възникване на събития (напр. може да изпрати уведомление в канал X при възникването на събитие 'Потребител се присъедини в сървъра').

Модулът разполага и с отделна подбиблиотека **discordjs/voice**, предвидена да осъществява взаимодействие специфично с гласовия API на Дискорд. Тя предоставя редица функции за работа с гласови канали, включително възможност за присъединяване и напускане на гласови канали и възпроизвеждане на различни видове аудио файлове.[17]

1.5.MySQL, SQL и mysql.js

MySQL е популярна, многопоточна, многопотребителска система за управление на релационни бази данни (СУРБД) с отворен код, която се използва за съхраняване, управление и извличане на данни. Тя бе въведена през 1995 г. и днес е една от най-използваните СУРБД в света, захранваща много уебсайтове и приложения.

Системата първоначално бе създадена като лека и бърза алтернатива на други СУРБД като Oracle и Microsoft SQL Server. Тя бързо придоби популярност сред уеб разработчиците и се превърна в система за база данни по подразбиране за много популярни приложения, вкл. WordPress, Drupal и Joomla. В допълнение към своята популярност в уеб приложенията, MySQL намира приложение и в широк спектър от други области, като банкиране, телекомуникации и научни изследвания.

MySQL е създаден от шведската компания MySQL AB, основана от Майкъл Видениус, Дейвид Аксмарк и Алън Ларсон през 1995 г. Тази система се разпространява под Общия публичен лиценз на GNU (*GNU General Public License*), което позволява свободното ѝ използване, модифициране и разпространяване. Предлага се и търговски лиценз за организации, които не желаят да разпространяват своите приложения под Общия публичен лиценз.[18]

SQL (Structured Query Language) е език за програмиране, използван за управление на релационни бази данни. Той позволява на програмистите да създават и актуализират бази данни и да изпращат заявки към тях, използвайки стандартизиран синтаксис, който е общ за много различни системи за управление на бази данни, включително MySQL. SQL се състои от набор от команди и изрази, които се използват за създаване и изтриване на таблици, вмъкване и актуализиране на данни, изпращане на заявки до базата данни за извличане на специфична информация и др.[19]

Mysql.js е модул на Node.js, играещ ролята на драйвер за MySQL. Този модул прави взаимодействието на системата за управление на релационни бази данни MySQL със средата за изпълнение Node.js възможно, ефективно и лесно. Mysql.js е написан изцяло на JavaScript и е безплатно достъпен през софтуерния регистър npm.

След успешно свързване на Node.js приложението си с база данни, програмистите имат възможността да изпращат заявки към нея, чрез които да извършват различни операции. Заявките се изпращат с помощта на метод `query()`, като текстът на заявката (написан на SQL език) се приема като задължителен аргумент на функцията. Разработчиците разполагат с пълния набор от команди, които SQL предлага.[20]

1.6.phpMyAdmin

phpMyAdmin е известен уеб базиран инструмент с отворен код, който се използва за управление на MySQL и MariaDB бази данни. Той предоставя удобен интерфейс, чрез който разработчиците лесно да създават, модифицират и изтриват бази данни, таблици и други обекти на БД, както и да управляват потребителски акаунти и изпълняват SQL заявки. Инструментът е написан на PHP и е безплатно достъпен през уеб браузър.[21]

1.7. XAMPP

XAMPP е безплатно, междуплатформено решение за уеб сървър с отворен код, което включва HTTP сървъра Apache, MySQL база данни, скриптовия език PHP и други инструменти, необходими за локално разработване и внедряване на уеб приложения. XAMPP е наличен за операционните системи Windows, Linux и macOS.

XAMPP означава **X** (междуплатформен), **A**pache, **M**ySQL, **P**HP и **P**erl. Пакетът включва всичко, от което разработчиците се нуждаят, за да настройат среда на локален уеб сървър, което им позволява да разработват и тестват своите уеб приложения на собствените си компютри, преди да ги разположат на главния сървър.

Решението осигурява лесен за използване графичен интерфейс, от който програмистите могат да стартират, спират и управляват различните компоненти на уеб сървъра, както и да конфигурират техните настройки. Това опростява процеса по настройка на локална среда за разработка.[22]

1.8. HTML

HTML (HyperText Markup Language) е език за маркиране, използван за създаване на структурата и съдържанието на уеб страниците. Той е ключов компонент на уеб разработката, заедно с CSS и JavaScript.

HTML предоставя набор от етикети (тагове) за маркиране, които описват структурата и съдържанието на уеб страница. Тези тагове включват заглавия, параграфи, списъци, изображения, връзки, таблици и много други. Използвайки тези тагове, програмистите създават йерархия от елементи, които определят структурата на страницата.

В допълнение към структурните елементи, езикът позволява на разработчиците да добавят и семантично значение към своите страници, използвайки специални етикети. Например етикетът „article“ може да се използва за маркиране на новинарска статия, докато „nav“ може да се използва за маркиране на навигационно меню.

HTML документите обикновено се пишат в редактор на код (напр. VS Code) или в обикновен текстов редактор, като се записват с файлово разширение „.html“. Те могат да се разглеждат в уеб браузър, който интерпретира етикетите за маркиране и показва съдържанието на страницата.

Като цяло HTML е фундаментална и стандартна технология за уеб разработка, осигуряваща основата за структурата и съдържанието на уеб страниците.[23]

1.9.CSS

CSS (Cascading Style Sheets) е език за описание на стилове, използван за външното представяне и оформление на уеб страниците. Работи съвместно с HTML и JavaScript за създаване на визуално привлекателни и отзивчиви уеб страници.

CSS позволява дефиниране на стилове за отделни HTML елементи или групи от елементи, като шрифтове, цветове, разстояние, оформление и т.н. Чрез отделяне на презентационния слой (CSS) от слоя на съдържанието (HTML), разработчиците могат лесно да правят промени в дизайна на уеб страница, без да засягат съдържанието.

Езикът използва набор от селектори, за да прилага стилове към конкретни HTML елементи. Например селекторът „h1“ е насочен към всички тагове "h1" на уеб страница, което позволява на разработчиците да дефинират един и същ стил за всички заглавия.

CSS може да бъде написан директно в HTML документ или във външен файл. Препоръчително е да се използват външни файлове при по-големи проекти, тъй като позволяват по-добра организация и по-лесна поддръжка.[24]

1.10. jQuery

jQuery е JavaScript библиотека с отворен код, която опростява процеса по манипулиране на HTML документи и обработка на събития на уеб страници. Библиотеката е създадена през 2006 г. и с времето е станала задължителна част от "кутията с инструменти" на много уеб разработчици.

Едно от основните предимства на jQuery е способността му да опрости процеса на избиране и манипулиране на HTML елементи в DOM. Програμισите използват простия синтаксис на jQuery, за да избират лесно елементи и да извършват действия върху тях.

Библиотеката съдържа редица вградени ефекти и анимации, които могат да бъдат приложени към HTML елементи за подобряване на визуалното изживяване на посетителите на уеб страницата.[25]

1.11. dotenv

Dotenv е модул на Node.js с нулев брой зависимости, който се използва за зареждане на променливи на средата (environment variables) от ".env" файл в Node.js приложение. Пакетът автоматично анализира файла и попълва стойностите на свойствата на обекта process.env, като по този начин ги прави достъпни в рамките на цялото приложение.

2. Архитектура на системата

Архитектурата на системата (**Фиг. 10**) се състои от следните елементи:

- **Потребители на Дискорд** – членовете на сървъра, в който се намира StudyBuddy;
- **Приложно-програмен интерфейс (API) и сървър на Дискорд** – поддържат се и се управляват от компанията Discord Inc;
- **Хост сървър на бота** – локален сървър (хостнат на персонален компютър) или официален хостван сървър в облака за работа 24 часа 7 дни в седмицата;
- **Програмен код на бота** – програмните инструкции, указващи начинът, по който StudyBuddy ще реагира на различни събития и интеракции (взаимодействия).

Процесът започва, когато потребител на Дискорд взаимодейства по някакъв начин с платформата (напр. изпращане на съобщение в текстови канал, кликване върху бутон, правене на избор от падащ списък и т.н.). След възникване на потребителско действие, ботът взаимодейства с API-то, за да установи точно какви събития и механизми са се задействали. След като това бъде установено, ботът реагира по начинът, по който е програмиран (напр. извежда, модифицира или изтрива съобщение, създава текстови канали и т.н.)



Фиг. 10: Архитектура на предложеното решение

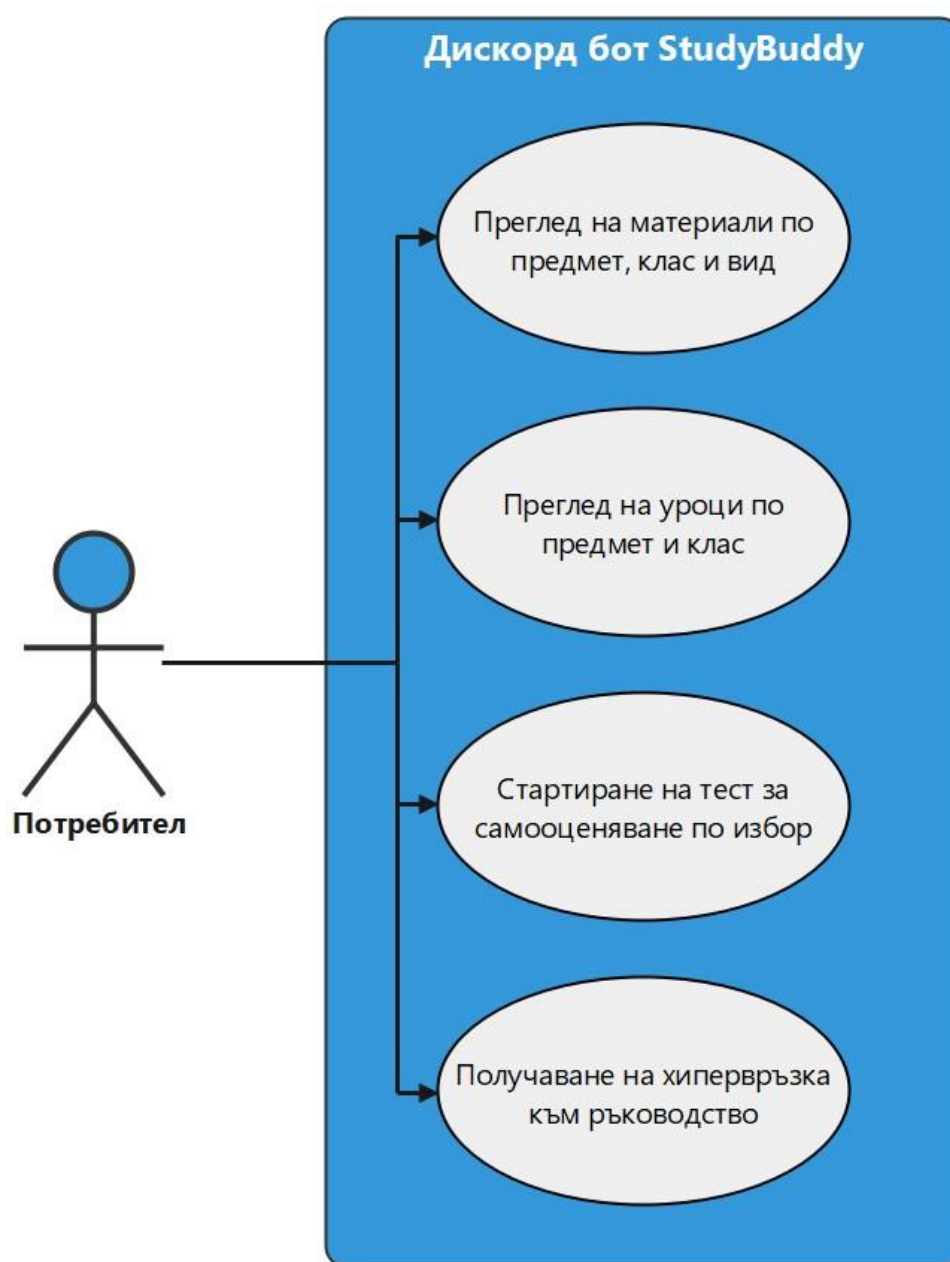
3. Логически модел на системата

3.1. Потребителски случаи

3.1.1. Диаграма на случаи на употреба

UML (Unified Modeling Language) е стандартизиран език за визуално моделиране, който обикновено се използва в софтуерното инженерство за описание и проектиране на софтуерни системи. Той предоставя набор от графични обозначения, които могат да се използват за представяне на различните елементи на система, като класове, обекти, компоненти и техните взаимоотношения и поведения.

Този унифициран език е използван, за да се моделира обща диаграма на потребителските случаи (**Фиг. 11**).



Фиг. 11: Обща диаграма на потребителските случаи
(general use-case diagram)

3.1.2. Сценарии на потребителски случаи

Сценарии за услугата „Преглед на материали по предмет, клас и вид“

Основен сценарий

1. Услугата стартира, когато потребителят изпрати командата „материали“ в текстов канал на сървър в Дискорд. Командата приема 3 задължителни аргумента: предмет, клас и вид материал.
2. Ботът проверява дали разполага с необходимите сървърни разрешения. Ако отговорът е не, се изпълнява алтернативен сценарий А1.
3. Ботът изпраща заявка към БД. Ако заявката е неуспешна, се изпълнява сценарий за грешка Е1.
4. Ботът проверява дали БД разполага с информацията, която потребителят заявява. Ако отговорът е не, се изпълнява алтернативен сценарий А2.
5. Ботът извежда меню (падащ списък), като указва, че чрез него трябва да бъде избран урок и подканва потребителя да направи своя избор.
6. Потребителят избира урок от падащия списък.
7. Ако потребителят е избрал „изображение“ като вид материал, ботът изпълнява алтернативен сценарий А3; ако потребителят е избрал „аудио“ – ботът изпълнява алтернативен сценарий А4.
8. Ботът извежда съобщение, съдържащо поисканата от потребителя информация, като запазва падащото меню за възможност за избор на друг урок.
9. Потребителят прочита информацията и разполага с ограничено време за избор на друг урок от менюто. Ако потребителят избере друг урок, се изпълнява алтернативен сценарий А5.
10. Времето за избор на урок изтече, ботът премахва всички менюта и изпраща уведомително съобщение в чата.
11. Услугата приключва.

Алтернативен сценарий А1 – Липса на задължителни сървърни разрешения

1. Ботът уведомява потребителя, че услугите му са недостъпни, тъй като не разполага с необходимите сървърни разрешения.
2. Ботът уведомява собственика на сървъра, в който потребителят е изпратил командата, че се нуждае от настройка, за да извършва дейността си.
3. Услугата приключва.

Алтернативен сценарий А2 – Липса на данни в БД, отговарящи на подадените от потребителя критерии

1. Ботът уведомява потребителя, че не разполага с никакви материали по зададените от него критерии.
2. Услугата приключва.

Алтернативен сценарий А3 – Опция „изображение“ като вид материал

1. Ботът изпълнява стъпка 8 от *Основен сценарий*, като под падащото меню за избор на урок добавя и падащо меню за избор на текущо изображение за преглед.
2. Потребителят разглежда текущото изображение и разполага с ограничено време за избор на друго изображение или друг урок от менютата. Ако потребителят избере друг урок или изображение, се изпълнява алтернативен сценарий А5.
3. Времето за избор на урок или изображение изтече, ботът премахва менютата и изпраща уведомително съобщение в чата.
4. Услугата приключва.

Алтернативен сценарий А4 – Опция „аудио“ като вид материал

1. Ботът извежда съобщение, съдържащо списък с наличните аудиоклипове от текущо избрания урок, като запазва падащото меню за възможност за избор на друг урок и добавя второ падащо меню за избор на аудиоклип за възпроизвеждане.
2. Ботът подканва потребителя да избере аудиоклип. Ако потребителят избере друг урок, се изпълнява алтернативен сценарий А5.
3. Потребителят избира аудиоклип за възпроизвеждане.
4. Ботът проверява дали потребителя се намира в гласови канал. Ако отговорът е не, се изпълнява алтернативен сценарий А6. Ако отговорът е да, ботът се присъединява към гласовия канал на потребителя, започва възпроизвеждане на аудиоклипа и създава „Стоп“ бутон.
5. Ботът извежда уведомително съобщение в чата.
6. След като ботът приключи с възпроизвеждането на аудиоклипа, той изчаква 30 секунди, след които напуска гласовия канал поради неактивност и изпраща уведомително съобщение в чата.
7. Времето за избор на урок или аудиоклип изтече, ботът премахва менютата и изпраща уведомително съобщение в чата.
8. Услугата приключва.

Алтернативен сценарий А5 – Промяна на текущо съдържание чрез правене на избор от падащ списък

1. Ботът променя съдържанието въз основа на новонаправения избор.
2. Ботът нулира таймера, като по този начин дава още време за избор/преглед на съдържание на потребителя.
3. Услугата приключва.

Алтернативен сценарий А6 – Потребителят не е присъединен към гласов канал

1. Ботът уведомява потребителя, че за възпроизвеждане на аудиоклипа е необходимо потребителя да се присъедини към гласов канал.
2. Услугата приключва.

Сценарий за грешка Е1 – Възникване на грешка при опит за изпълнение на заявка към базата данни

1. Ботът уведомява потребителя, че в момента услугите му са недостъпни и го моли да опита отново по-късно.
2. Ботът извежда текста на грешката в терминала.
3. Услугата приключва.

Сценарии за услугата „Преглед на уроци по предмет и клас“

Основен сценарий

1. Услугата стартира, когато потребителят изпрати командата „уроци“ в текстов канал на сървър в Дискорд. Командата приема 2 задължителни аргумента: предмет и клас.
2. Ботът проверява дали разполага с необходимите сървърни разрешения. Ако отговорът е не, се изпълнява алтернативен сценарий А1.
3. Ботът изпраща заявка към БД. Ако заявката е неуспешна, се изпълнява сценарий за грешка Е1.
4. Ботът проверява дали БД разполага с информацията, която потребителят заявява. Ако отговорът е не, се изпълнява алтернативен сценарий А2.
5. Ботът извежда меню (падащ списък), като указва, че чрез него трябва да бъде избран урок и подканва потребителя да направи своя избор.
6. Потребителят избира урок от падащия списък.
7. Ботът извежда съобщение, съдържащо всички налични материали (текст, изображения, видео и аудио), свързани с избора на урок, като запазва падащото меню за възможност за избор на друг урок.

8. Потребителят прочита информацията и разполага с ограничено време за избор на друг урок от менюто. Ако потребителят избере друг урок, се изпълнява алтернативен сценарий А3.
9. Времето за избор на урок изтече, ботът премахва всички менюта и изпраща уведомително съобщение в чата.
10. Услугата приключва.

Алтернативен сценарий А1 – Липса на задължителни сървърни разрешения

1. Ботът уведомява потребителя, че услугите му са недостъпни, тъй като не разполага с необходимите сървърни разрешения.
2. Ботът уведомява собственика на сървъра, в който потребителят е изпратил командата, че се нуждае от настройка, за да извършва дейността си.
3. Услугата приключва.

Алтернативен сценарий А2 – Липса на данни в БД, отговарящи на подадените от потребителя критерии

1. Ботът уведомява потребителя, че не разполага с никакви материали по зададените от него критерии.
2. Услугата приключва.

Алтернативен сценарий А3 – Промяна на текущо съдържание чрез правене на избор от падащ списък

1. Ботът променя съдържанието въз основа на новонаправения избор.
2. Ботът нулира таймера, като по този начин дава още време за избор/преглед на съдържание на потребителя.
3. Услугата приключва.

Сценарий за грешка Е1 – Възникване на грешка при опит за изпълнение на заявка към базата данни

1. Ботът уведомява потребителя, че в момента услугите му са недостъпни и го моли да опита отново по-късно.
2. Ботът извежда текста на грешката в терминала.
3. Услугата приключва.

Сценарии за услугата „Стартиране на тест за самооценяване по избор“

Основен сценарий

1. Услугата стартира, когато потребителят изпрати командата „тестове“ в текстов канал на сървър в Дискорд. Командата приема 2 задължителни аргумента: предмет и клас.
2. Ботът проверява дали разполага с необходимите сървърни разрешения. Ако отговорът е не, се изпълнява алтернативен сценарий А1.
3. Ботът изпраща заявка към БД. Ако заявката е неуспешна, се изпълнява сценарий за грешка Е1.
4. Ботът проверява дали БД разполага с информацията, която потребителят заявява. Ако отговорът е не, се изпълнява алтернативен сценарий А2.
5. Ботът извежда меню (падащ списък), като указва, че чрез него трябва да бъде избран тест и подканва потребителя да направи своя избор.
6. Потребителят избира тест от падащия списък.
7. Ботът извежда съобщение, чрез което уведомява потребителя, че теста ще стартира след 30 секунди и започва обратно броене. Съобщението съдържа и информация, описваща характеристиките на избрания от потребителя тест (предмет, клас, брой въпроси, време за изпълнение).
8. Потребителят прочита съобщението и търпеливо изчаква началото на теста.
9. Ботът визуализира първия въпрос от теста, възможните отговори, две менюта (едно за избор на въпрос за преглед и едно за избор на отговор на въпрос), както и брояч на оставащо време за изпълнение на теста.
10. Потребителят отговаря на всички въпроси или времето за изпълнение на теста свършва.
11. Ботът визуализира резултатите от теста: оценка, брой правилно отговорени въпроси, брой отговорени въпроси, общ брой въпроси, отнето време за изпълнение, както и списък с всички въпроси, посочен от потребителя отговор и правилен отговор.
12. Услугата приключва.

Алтернативен сценарий А1 – Липса на задължителни сървърни разрешения

1. Ботът уведомява потребителя, че услугите му са недостъпни, тъй като не разполага с необходимите сървърни разрешения.
2. Ботът уведомява собственика на сървъра, в който потребителят е изпратил командата, че се нуждае от настройка, за да извършва дейността си.
3. Услугата приключва.

Алтернативен сценарий А2 – Липса на данни в БД, отговарящи на подадените от потребителя критерии

1. Ботът уведомява потребителя, че не разполага с никакви материали по зададените от него критерии.
2. Услугата приключва.

Сценарий за грешка Е1 – Възникване на грешка при опит за изпълнение на заявка към базата данни

1. Ботът уведомява потребителя, че в момента услугите му са недостъпни и го моли да опита отново по-късно.
2. Ботът извежда текста на грешката в терминала.
3. Услугата приключва.

Сценарии за услугата „Получаване на хипервръзка към ръководство“
Основен сценарий

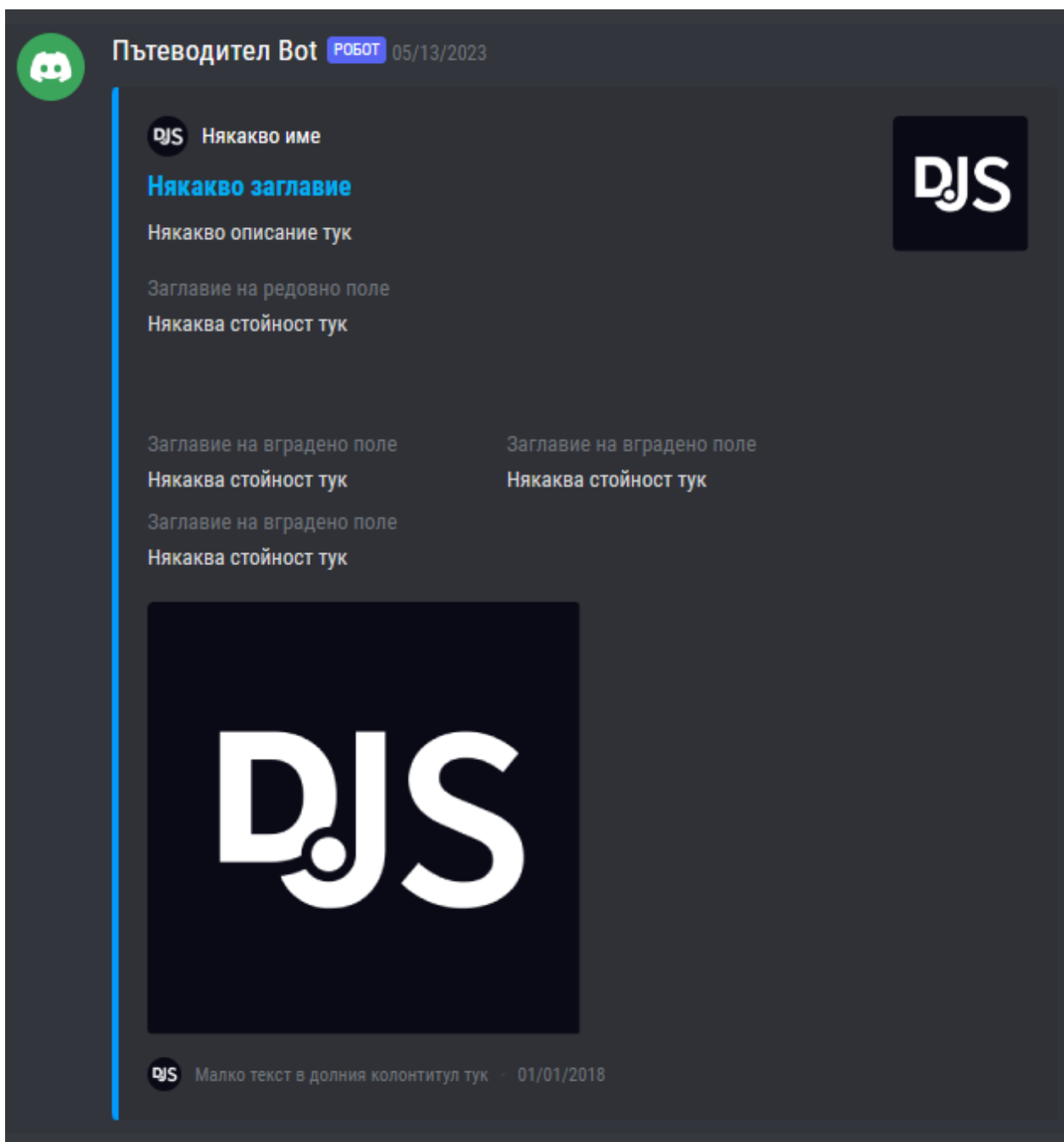
1. Услугата стартира, когато потребителят изпрати командата „ръководство“ в текстов канал на сървър в Дискорд.
2. Ботът извежда съобщение, съдържащо линк към ръководство в уебсайта на StudyBuddy.
3. Потребителят щраква върху линка.
4. Ръководството се визуализира на екрана на потребителя.
5. Услугата приключва.

4. Описание на потребителските интерфейси

Съдържанието, което може да бъде визуализирано от StudyBuddy, се представя чрез т.нар. ембеди (embeds, вградено съобщение).

В Дискорд „embed“ представлява специален формат на съобщение, който позволява на потребителите да създават богати, интерактивни съобщения с персонализирано форматиране, изображения, връзки и т.н. Когато потребител изпрати вградено съобщение, то се появява в чата като стилизиран интерфейс, съдържащ съдържанието, което потребителят е добавил (**Фиг. 12**).

На **Фиг. 13.1**, **Фиг. 13.2** и **Фиг. 13.3** са изобразени някои от интерфейсите, които потребителят вижда при употребата на различни команди (и аргументи към тях) на StudyBuddy.



Фиг. 12: Демонстрация на възможностите на embed в Дискорд [27]

Глава 3. Описание на проекта на приложението

1. Описание на базата данни

С цел динамичност на информацията и избягване на нужда от намеса от програмист, StudyBuddy е проектиран да използва MySQL база от данни като източник на информация, вместо локални файлове (напр. .json файлове). Благодарение на това решение, процесът по добавяне на нови материали и тестове и модифициране на вече добавените не изисква знания по програмиране.

За реализирането на приложението е използвана релационна база данни MySQL, като управлението ѝ се извършва с помощта на XAMPP и phpMyAdmin.

1.1. Описание на таблиците в БД

Базата данни съдържа 10 таблици (**Фиг. 14**), като данните са нормализирани.

Таблица	Действие	Редове	Тип	Колация	Размер
<input type="checkbox"/> answer	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	13	InnoDB	utf8_general_ci	32.0 КБ
<input type="checkbox"/> exam	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	1	InnoDB	utf8_general_ci	32.0 КБ
<input type="checkbox"/> exam_question_answer	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	21	InnoDB	utf8_general_ci	48.0 КБ
<input type="checkbox"/> grade	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	3	InnoDB	utf8_general_ci	16.0 КБ
<input type="checkbox"/> lesson	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	27	InnoDB	utf8_general_ci	32.0 КБ
<input type="checkbox"/> lesson_content	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	142	InnoDB	utf8_general_ci	272.0 КБ
<input type="checkbox"/> lesson_content_type	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	4	InnoDB	utf8_general_ci	16.0 КБ
<input type="checkbox"/> question	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	7	InnoDB	utf8_general_ci	16.0 КБ
<input type="checkbox"/> question_answer	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	21	InnoDB	utf8_general_ci	48.0 КБ
<input type="checkbox"/> subject	★ Прелистване Структура Търсене Вмъкване Изчистване Изтриване	9	InnoDB	utf8_general_ci	32.0 КБ
10 таблици	Сума	248	InnoDB	utf8_general_ci	544.0 КБ

Фиг. 14: Списък с всички таблици в базата данни

1.1.1. Таблица „grade“

Използва се за съхранение на възможните училищни класове. Структурата на таблицата (**Фиг. 14.1**) има две полета: **id** (идентификатор) и **grade** (клас). **Фиг. 14.1.1** показва въведените записи в таблицата.

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id 🔑	int(10)			Не	Няма		AUTO_INCREMENT
2	grade	int(2)			Не	Няма		

Фиг. 14.1: Структура на таблицата „grade“

				id	grade
<input type="checkbox"/>	Edit	Copy	Delete	1	5
<input type="checkbox"/>	Edit	Copy	Delete	2	6
<input type="checkbox"/>	Edit	Copy	Delete	3	7

Фиг. 14.1.1: Таблица „grade“

1.1.2. Таблица „subject“

Използва се за съхранение на възможните учебни предмети. Структурата на таблицата (Фиг. 14.2) има три полета: **id** (идентификатор), **name** (име на предмет) и **grade_id** (външен ключ за връзка с таблицата grade). Фиг. 14.2.1 показва въведените записи в таблицата.

#	Име	Тип	Колагия	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id	int(10)			Не	Няма		AUTO_INCREMENT
2	name	varchar(50)	utf8_general_ci		Не	Няма		
3	grade_id	int(10)			Не	Няма		

Фиг. 14.2: Структура на таблицата „subject“

				id	name	grade_id
<input type="checkbox"/>	Edit	Copy	Delete	1	География и икономика	1
<input type="checkbox"/>	Edit	Copy	Delete	2	История и цивилизации	1
<input type="checkbox"/>	Edit	Copy	Delete	3	Информационни технологии	1
<input type="checkbox"/>	Edit	Copy	Delete	4	География и икономика	2
<input type="checkbox"/>	Edit	Copy	Delete	5	История и цивилизации	2
<input type="checkbox"/>	Edit	Copy	Delete	6	Информационни технологии	2
<input type="checkbox"/>	Edit	Copy	Delete	7	География и икономика	3
<input type="checkbox"/>	Edit	Copy	Delete	8	История и цивилизации	3
<input type="checkbox"/>	Edit	Copy	Delete	9	Информационни технологии	3

Фиг. 14.2.1: Таблица „subject“

1.1.3. Таблица „lesson“

Използва се за съхранение на възможните учебни уроци. Структурата на таблицата (Фиг. 14.3) има четири полета: **id** (идентификатор), **number** (номер на урок), **name** (име на урок), **subject_id** (външен ключ за връзка с таблицата subject). Фиг. 14.3.1 показва някои от въведените записи в таблицата.

#	Име	Тип	Колагия	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id 🔑	int(10)			Не	Няма		AUTO_INCREMENT
2	number	int(2)			Не	Няма		
3	name	varchar(100)	utf8_general_ci		Не	Няма		
4	subject_id 🔑	int(10)			Не	Няма		

Фиг. 14.3: Структура на таблицата „lesson“

id	number	name	subject_id
1	1	Географията - древна и съвременна наука	1
2	2	Ориентиране в природата	1
3	3	Географски глобус и географска карта	1
4	1	Географско положение, големина, брегове и опознаване на Южна Америка	4
5	2	Релеф и полезни изкопаеми	4

Фиг. 14.3.1: Таблица „lesson“

1.1.4. Таблица „lesson_content_type“

Използва се за съхранение на възможните типове съдържание на урок. Структурата на таблицата (Фиг. 14.4) има две полета: **id** (идентификатор) и **type** (вид). Фиг. 14.4.1 показва въведените записи в таблицата.

#	Име	Тип	Колагия	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id 🔑	int(10)			Не	Няма		AUTO_INCREMENT
2	type	varchar(20)	utf8_general_ci		Не	Няма		

Фиг. 14.4: Структура на таблицата „lesson_content_type“

	id	type
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Текст
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Изображение
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Видеоклип
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Аудиоклип

Фиг. 14.4.1: Таблица „lesson_content_type“

1.1.5. Таблица „lesson_content“

Използва се за съхранение на съдържанието на урок. Структурата на таблицата (Фиг.14.5) има пет полета: **id** (идентификатор), **content** (съдържание), **content_description** (описание на съдържанието), **content_type_id** (външен ключ за

връзка с таблицата lesson_content_type) и **lesson_id** (външен ключ за връзка с таблицата lesson). **Фиг. 14.5.1** показва някои от въведените записи в таблицата.

StudyBuddy черпи информация основно от тази таблица, тъй като тук се съхранява текста на уроците, изображенията (линкове към тях и тяхното описание), видеоклиповете (линкове към тях и тяхното описание) и аудиоклиповете (техния път в локалната директория и тяхното описание).

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id 🔑	int(10)			Не	Няма		AUTO_INCREMENT
2	content	varchar(4090)	utf8_general_ci		Не	Няма		
3	content_description	varchar(256)	utf8_general_ci		Да	NULL		
4	content_type_id 🔑	int(10)			Не	Няма		
5	lesson_id 🔑	int(10)			Не	Няма		

Фиг. 14.5: Структура на таблицата „lesson_content“

id	content	content_description	content_type_id	lesson_id
1	**Науката география.** Думата география произлиза ...	NULL	1	1
2	https://i.imgur.com/Bjiwa1F.jpg	Фиг. 1: Географията - наука за природата и обществ...	2	1
3	https://i.imgur.com/HuuBB0a.jpg	Фиг. 2: Велики географски открития	2	1
4	https://i.imgur.com/A2UgvIP.jpg	NULL	2	1
5	**Хоризонт и посоки на хоризонта.** Повърхността н...	NULL	1	2

Фиг. 14.5.1: Таблица „lesson_content“

1.1.6. Таблица „question“

Използва се за съхранение на възможните въпроси в тест. Структурата на таблицата (**Фиг. 14.6**) има две полета: **id** (идентификатор) и **question** (въпрос). **Фиг. 14.6.1** показва въведените записи в таблицата.

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id 🔑	int(10)			Не	Няма		AUTO_INCREMENT
2	question	varchar(255)	utf8_general_ci		Не	Няма		

Фиг. 14.6: Структура на таблицата „question“

	id	question
<input type="checkbox"/> Edit Copy Delete	1	С какво е известен Христофор Колумб?
<input type="checkbox"/> Edit Copy Delete	2	С какво е известен Вашку да Гама?
<input type="checkbox"/> Edit Copy Delete	3	С какво е известен Фернандо Магелан?
<input type="checkbox"/> Edit Copy Delete	4	Какво изучава географията?
<input type="checkbox"/> Edit Copy Delete	5	Какво е хоризонт?
<input type="checkbox"/> Edit Copy Delete	6	Кои са методите за ориентиране?
<input type="checkbox"/> Edit Copy Delete	7	Какво е линия на хоризонта?

Фиг. 14.6.1: Таблица „question“

1.1.7. Таблица „answer“

Използва се за съхранение на възможните отговори в тест. Структурата на таблицата (Фиг. 14.7) има две полета: **id** (идентификатор) и **answer** (отговор). Фиг. 14.7.1 показва някои от въведените записи в таблицата.

#	Име	Тип	Колагия	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id	int(10)			Не	Няма		AUTO_INCREMENT
2	answer	varchar(255)	utf8_general_ci		Не	Няма		




Фиг. 14.7: Структура на таблицата „answer“

	id	answer
<input type="checkbox"/> Edit Copy Delete	1	Не е известен.
<input type="checkbox"/> Edit Copy Delete	2	С откриването на Америка.
<input type="checkbox"/> Edit Copy Delete	3	С откриването на морски път до Индия.
<input type="checkbox"/> Edit Copy Delete	4	С извършването на първото околосветско плаване.
<input type="checkbox"/> Edit Copy Delete	5	Природни обекти (планини, реки, гори и др.) и обек...

Фиг. 14.7.1: Таблица „answer“

1.1.8. Таблица „question_answer“

Използва се за свързваща таблица (junction table) на таблиците question и answer. Структурата на таблицата (Фиг. 14.8) има четири полета: **id** (идентификатор), **question_id** (външен ключ за връзка с таблицата question), **answer_id** (външен ключ за връзка с таблицата answer) и **is_correct** (използва се за указване на правилен отговор). Фиг. 14.8.1 показва някои от въведените записи в таблицата.

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id 	int(10)			Не	Няма		AUTO_INCREMENT
2	question_id 	int(10)			Не	Няма		
3	answer_id 	int(10)			Не	Няма		
4	is_correct	tinyint(1)			Не	0		



Фиг. 14.8: Структура на таблицата „question_answer”

id	question_id	answer_id	is_correct
1	1	2	1
2	1	3	0
3	1	4	0
4	2	4	0
5	2	2	0






Фиг. 14.8.1: Таблица „question_answer”

1.1.9. Таблица „exam”

Използва се за съхранение на възможните тестове. Структурата на таблицата (Фиг. 14.9) има две полета: **id** (идентификатор) и **subject_id** (външен ключ за връзка с таблицата subject). Фиг. 14.9.1 показва въведените записи в таблицата.

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително
1	id 	int(10)			Не	Няма		AUTO_INCREMENT
2	subject_id 	int(10)			Не	Няма		

Фиг. 14.9: Структура на таблицата „exam”

 T 	▼	id	subject_id
<input type="checkbox"/>  Edit  Copy  Delete		1	1

Фиг. 14.9.1: Таблица „exam”

1.1.10. Таблица „exam_question_answer”

Използва се за свързваща таблица (junction table) на таблиците question_answer и exam. Структурата на таблицата (Фиг. 14.10) има две полета: **exam_id** (външен ключ за връзка с таблицата exam) и **qa_id** (външен ключ за връзка с таблицата question_answer). Фиг. 14.10.1 показва някои от въведените записи в таблицата.

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнителни
1	exam_id 🔑 🔒	int(10)			Не	Няма		
2	qa_id 🔑 🔒	int(10)			Не	Няма		

Фиг. 14.10: Структура на таблицата „exam_question_answer”

← T →	exam_id	qa_id
<input type="checkbox"/> Edit Copy Delete	1	1
<input type="checkbox"/> Edit Copy Delete	1	2
<input type="checkbox"/> Edit Copy Delete	1	3
<input type="checkbox"/> Edit Copy Delete	1	4
<input type="checkbox"/> Edit Copy Delete	1	5

Фиг. 14.10.1: Таблица „exam_question_answer”

Нови уроци и тестове могат да бъдат въвеждани в базата данни по всяко време. Към този момент е налична толкова информация, колкото е необходима за демонстриране работата на предложеното решение.

Източниците на учебните материали, използвани за реализацията на този проект, са следните (конвенция APA):

➤ География и икономика

- 5 клас – Попов, А. Б., Константинова, Е., Милкова, К. Д., Ангелова, Д. П., & Здравкова, Ж. В. (2019). *География и икономика: пети клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/619> (Оригинал публикуван 2016)
- 6 клас – Попов, А. Б., Ангелова, Д. П., & Любенова, Л. И. (2019). *География и икономика: шести клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/641> (Оригинал публикуван 2017)
- 7 клас - Попов, А. Б., Найденов, К. М., Ангелова, Д. П., & Любенова, Л. И. (2018). *География и икономика: седми клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/418>

➤ История и цивилизации

- 5 клас – Матанов, Х. Л., Делев, П. Н., Каснакова, Ц. И., Мукарзел, К. С., & Рангелова, Т. П. (2019). *История и цивилизации: пети клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/601>
- 6 клас – Матанов, Х. Л., Димитров, Т. О., & Иванова, Г. Ю. (2019). *История и цивилизации: шести клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/642> (Оригинал публикуван 2017)

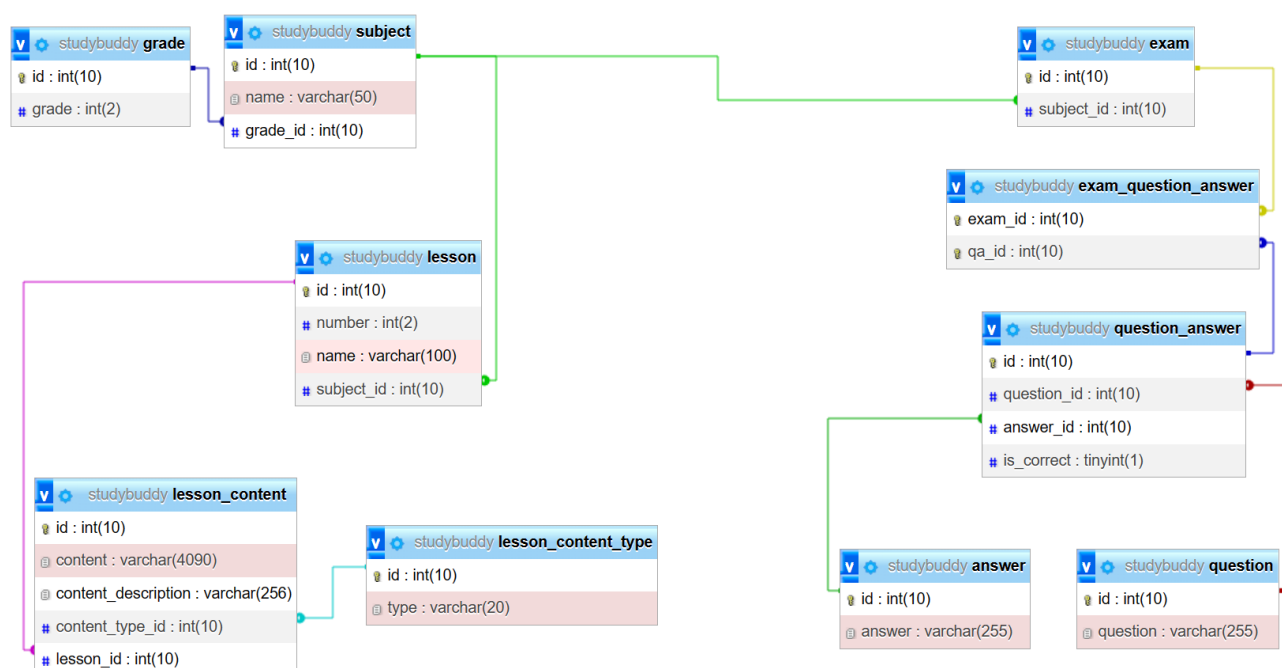
- 7 клас – Матанов, Х. Л., Колев, В. К., Маринова-Христиди, Р. Д., & Мишев, Т. И. (2018). *История и цивилизации: седми клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/416>

➤ Информационни технологии

- 5 клас – Петров, В. И., Дженкова, З. С., Пенчева, К. Т., & Зангочева И. П. (2019). *Информационни технологии: пети клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/623> (Оригинал публикуван 2016)
- 6 клас – Петров, В. И., Пенчева, К. Т., & Георгиева, Н. Й. (2019). *Информационни технологии: шести клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/640> (Оригинал публикуван 2017)
- 7 клас – Петров, В. И., Георгиева, Н. Й., & Тодорова, И. В. (2018). *Информационни технологии: седми клас*. Клет България, Анубис. <https://sales.anubis-bulvest.com/parents/useBook/415>

1.2. Релационен модел на БД

Релационният модел на базата данни (**Фиг. 15**) показва обективно връзките между различните таблици.



Фиг. 15: Релационен модел на базата данни на StudyBuddy

2. Реализация на проекта

2.1. Реализация на приложението StudyBuddy

2.1.1. Структура

Главната директория на проекта на приложението съдържа следните файлове и папки:

- **index.js** – главен изпълним файл на приложението; изпълнява се при началното стартиране. Тук ботът бива удостоверяван в системата на Дискорд. Четат и обработват се всички команди от папката с команди, след което listener (слушател) „слуша“ за възникнали взаимодействия с приложението. Ако взаимодействието съвпада с налична команда, приложението изпълнява изпълнимия файл на съответната команда;
- **deploy-commands.js** – изпълним файл, създаден за динамично добавяне на наличните команди към списъка с команди на всеки Дискорд сървър, в който ботът членува;
- **package.json** и **package-lock.json** – конфигурационни файлове, автоматично генерирани при инициализацията на пакетния мениджър npm в директорията на проекта;
- **.env** – конфигурационен файл, съдържащ променливите на средата (**environment variables**). Съдържа редове чист текст, като всеки ред е двойка ключ-стойност, която представлява променлива. Файлът е използван за съдържание на жизненоважни променливи като токен на бота, данни за връзка с БД и т.н.;
- **commands** – папка, съдържаща изпълнимите файлове на всяка команда:
 - **materials.js** – изпълним файл на командата /материали;
 - **lessons.js** – изпълним файл на командата /уроци;
 - **exams.js** – изпълним файл на командата /тестове;
 - **guide.js** – изпълним файл на командата /ръководство;
- **functions** – папка, съдържаща изпълними файлове на различни потребителски дефинирани модули:
 - **hasPermissions.js** – изпълним файл на модул за проверка за наличие на необх. разрешения;
 - **Pagination.js** – изпълним файл на модул, съдържащ 4 функции, всяка от които описва обработката и визуализацията на различните видове материали от команда /материали;
 - **stopCountdown.js** – изпълним файл на модул за спиране на таймер;

- **database** – папка, съдържаща две подпапки: **audio** и **images**. Папката **audio** съдържа аудиофайлове, а **images** – изображения;
- **node_modules** – папка, автоматично генерирана при инициализацията на пакетния мениджър **npm** в директорията на проекта. Съдържа множество подпапки, всяка от които е отделена за всеки **node.js** модул.

2.1.2. Програмна реализация на основата (index.js)

```
// Зареждане на environment variables в приложението
require('dotenv').config();
// Изискване на нужни класове от discord.js
const { Client, Collection, Events, GatewayIntentBits } =
require('discord.js');
// Изискване на модули, необходими за четене на файловете за всяка команда
const fs = require('node:fs');
const path = require('node:path');
// Създаване на обект от клас Client
const client = new Client({ intents: [GatewayIntentBits.Guilds,
GatewayIntentBits.GuildMessages, GatewayIntentBits.MessageContent,
GatewayIntentBits.GuildVoiceStates] });
client.commands = new Collection();

// Изпращане на потвърждение в терминала при успешно стартиране на бота
client.once(Events.ClientReady, () => {
  console.log('Ботът бе стартиран успешно.');
```

```
});
// Обработка на всички команди
const commandsPath = path.join(__dirname, 'commands');
const commandFiles = fs.readdirSync(commandsPath).filter(file =>
file.endsWith('.js'));

for (const file of commandFiles) {
  const filePath = path.join(commandsPath, file);
  const command = require(filePath);

  if ('data' in command && 'execute' in command) {
    client.commands.set(command.data.name, command);
  } else {
    console.log(`[ПРЕДУПРЕЖДЕНИЕ] Липсва свойство 'data' или 'execute' в
следната команда: ${filePath}`);
  }
}

// Listener за команди
client.on(Events.InteractionCreate, async interaction => {
  if (!interaction.isChatInputCommand()) return; // Ако възникналият
interaction не е команден, събитието се пропуска

  const command = interaction.client.commands.get(interaction.commandName);
```

```

    // Ако командата не съществува се извежда грешка в терминала и събитието
    се пропуска
    if (!command) {
        console.error(`Командата '${interaction.commandName}' не
съществува.`);
        return;
    }

    try {
        // Изпълнение на командата
        await command.execute(interaction);
    } catch (err) {
        // Извеждане на грешката в терминала при възникване на такава и
        уведомяване на потребителя, въвел командата
        console.error(err);
        if (interaction.replied || interaction.deferred) {
            await interaction.followUp({ content: 'Възникна грешка при
изпълнението на тази команда. Моля, опитай отново!', ephemeral: true });
        } else {
            await interaction.reply({ content: 'Възникна грешка при
изпълнението на тази команда. Моля, опитай отново!', ephemeral: true });
        }
    }
});

// Логване на бота в Дискорд
client.login(process.env.BOT_TOKEN);

```

Фиг. 16: Програмен код на index.js

Може да се каже, че **index.js** е „сърцето“ на приложението (**Фиг. 16**). В този файл се указва начина, по който ботът реагира при възникването на взаимодействие с него.

Тук е мястото, където се зареждат променливите на средата от файла **.env**. Те са достъпни за всяка една част от приложението. Файлът **.env** (**Фиг. 17**) съдържа токена на бота, идентификатор на клиент и сървър за тестване на разработки, данни за база данни (име на хост, потребителско име, парола, име на БД), както и променлива за съхранение на често използвания цвят (цветът на StudyBuddy).

```

# Fundamental
BOT_TOKEN=MTA4OTg3MDE3NDA3ODkwMjMyMg.GN4Nvp.CoY0JmEwM2DUkdoEIiAV1lw
clientId=108987013463323322
devGuildId=70167634532397840

# Database
host=localhost
user=root
password=
db=studybuddy

# Design
embedColor=Blue

```

Фиг. 17: Текст на .env файла

В **index.js** се дефинира клиента (**client**), който всъщност е самия бот в API-то на Дискорд. При дефиницията му се указват намеренията (т.нар. **intents**) на приложението,

които указват кои събития получава приложението от API-то, въз основа на това каква функционалност се желае от него.

В този файл се осъществява динамично събиране и обработка на колекция от налични команди (намиращи се в папка `commands`). След като всички команди са събрани в колекция, с помощта на т.нар. слушател (`listener`) ботът отговаря на изпратените от потребител команди (ако такава съществува), като задейства съответния изпълним файл на командата. В случай на неуспешно изпълнение на команда, ботът ще уведоми потребителя за това.

2.1.3. Програмна реализация на връзката с БД и изпращане на SQL заявки

Безпроблемната връзка с базата данни е от съществено значение за приложението. Без нея, StudyBuddy не може да функционира изцяло.

Програмно, връзката с БД се осъществява с функцията **`createPool()`** на модула `mysql` (**Фиг. 18**). Този програмен код е имплементиран в изпълнимия файл на всяка команда, чиято функционалност изисква информация от базата (`materials.js`, `lessons.js`, `exams.js`).

```
// Създаване на връзка с базата данни
let mysql = require('mysql');
const pool = mysql.createPool({
  connectionLimit: 100,
  host: process.env.host,
  user: process.env.user,
  password: process.env.password,
  database: process.env.db,
  debug: false
});
```

Фиг. 18: Програмен код за връзка с БД

Достъпването на информацията в таблиците на БД се случва чрез т.нар. заявки (`queries`). Програмно, изпращането на заявки се реализира с помощта на функцията **`query()`**. На **Фиг. 19** е показан програмен код, чрез който се изпраща заявка към БД, проверява се за грешки при изпълнението ѝ и се извежда уведомително съобщение в случай на липса на данни. **Той е общ за всяка команда, при която се изпращат заявки.**

```

//Изпълнение на заявката
pool.query(query, async (error, queryResult) => {
    // При възникване на грешка при изпълнението на заявката
    (напр. неуспешна връзка с БД), ботът уведомява потребителите за
    неспособността си и извежда грешката в конзолата
    if (error) {
        interaction.reply('В момента услугите ми са недостъпни.
Моля, опитай по-късно!');
        return console.error(error);
    }

    // При липса на данни в БД, ботът указва това на потребителя
    чрез съобщение
    if (Object.keys(queryResult).length === 0) return
interaction.reply('Упс! Не разполагам с никакви материали по зададените
от теб критерии. :pensive:\nМоля, опитай с различна комбинация!
:relaxed:');
});

```

Фиг. 19: Програмен код за изпращане на заявки към БД

От програмния код (показан на Фиг. 19) става ясно, че функцията query() приема задължителен аргумент. Този аргумент указва текста на заявката, написан на езика SQL. На **Фиг. 20.1**, **Фиг. 20.2** и **Фиг. 20.3** е изобразен текста на SQL заявките, използвани при реализацията на командите /материали, /уроци и /тестове.

```

const query = `
    SELECT lesson.number, lesson.name, lesson_content.content,
    lesson_content_type.type, lesson_content.content_description
    FROM lesson_content
    INNER JOIN lesson_content_type ON lesson_content.content_type_id =
    lesson_content_type.id
    INNER JOIN lesson ON lesson_content.lesson_id = lesson.id
    INNER JOIN subject ON lesson.subject_id = subject.id
    INNER JOIN grade ON subject.grade_id = grade.id
    WHERE subject.name = '${subject}' AND grade.grade = ${grade} AND
    lesson_content_type.type = '${type}'
    ORDER BY lesson.number ASC, lesson_content.id ASC`;

```

Фиг. 20.1: SQL текст на заявка, използвана при команда /материали

```

const query = `
    SELECT lesson.number, lesson.name, lesson_content.content,
    lesson_content_type.type, lesson_content.content_description
    FROM lesson_content
    INNER JOIN lesson_content_type ON lesson_content.content_type_id =
    lesson_content_type.id
    INNER JOIN lesson ON lesson_content.lesson_id = lesson.id
    INNER JOIN subject ON lesson.subject_id = subject.id
    INNER JOIN grade ON subject.grade_id = grade.id
    WHERE subject.name = '${subject}' AND grade.grade = ${grade}
    ORDER BY lesson_content_type.id ASC, lesson_content.id ASC`;

```

Фиг. 20.2: SQL текст на заявка, използвана при команда /уроци

```

const query = `
  SELECT exam.id, question_answer.question_id, question.question,
  answer.answer, question_answer.is_correct
  FROM exam_question_answer
  INNER JOIN question_answer ON exam_question_answer.qa_id =
  question_answer.id
  INNER JOIN exam ON exam_question_answer.exam_id = exam.id
  INNER JOIN question ON question_answer.question_id = question.id
  INNER JOIN answer ON question_answer.answer_id = answer.id
  INNER JOIN subject ON exam.subject_id = subject.id
  INNER JOIN grade ON subject.grade_id = grade.id
  WHERE subject.name = '${subject}' AND grade.grade = ${grade}
  ORDER BY question_answer.question_id ASC`;

```

Фиг. 20.3: SQL текст на заявка, използвана при команда /тестове

2.1.4. Програмна реализация на командите

Всеки изпълним файл на команда съдържа в себе си атрибут **data** и функция **execute()** (Фиг. 21). Атрибутът data указва свойствата на командата. Те включват име, описание и различни видове опции за вход (аргументи). Всеки аргумент има свои свойства, като име, описание, възможни избори и др. Функцията execute() съдържа реалните операции, които ще се извършат при изпълнението на командата.

```

data: new SlashCommandBuilder()
  .setName('уроци')
  .setDescription('Изпраща всички налични материали по конкретния
предмет, клас и урок.')
  .addStringOption(subject =>
    subject
    .setName('предмет')
    .setDescription('Учебният предмет, по-който търсиш
материали.')
    .setRequired(true)
    .addChoices(
      { name: 'география и икономика', value: 'География и
икономика' },
      { name: 'история и цивилизации', value: 'История и
цивилизации' },
      { name: 'информационни технологии', value:
'Информационни технологии' },
    )
  )
  .addIntegerOption(grade =>
    grade
    .setName('клас')
    .setDescription('Класът, който си в момента.')
    .setRequired(true)
    .addChoices(
      { name: '5', value: 5 },
      { name: '6', value: 6 },
      { name: '7', value: 7 },
    )
  ),
  async execute(interaction) { // Код за изпълнение }

```

Фиг. 21: Структура на команда

Командите, чиято функционалност изисква допълнителни входни данни след първоначалното въвеждане, са три: **/материали**, **/уроци** и **/тестове**. Те са командите, които предоставят интерактивност с приложението. След първоначалното изпращане на някоя от тези команди, комуникацията между потребителя и StudyBuddy се извършва с помощта на т.нар. **SelectMenu**.

SelectMenu е компонент на интерфейса на Дискорд, представляващ падащ списък с опции, които потребител може да избере чрез щракване с мишката (**Фиг. 22**). Този компонент е използван на множество места при разработката на приложението, като най-често е употребяван като метод за навигация между различните изгледи. SelectMenu може да приема различни свойства, като персонализиран идентификатор, заглавие по подразбиране и др.

```
const selectLesson = new StringSelectMenuBuilder()
    .setCustomId('selectLesson')
    .setPlaceholder('Избери урок от тук')
for (let i = Object.keys(groupedQuery)[0]; i <=
Object.keys(groupedQuery).length; i++) {
    selectLesson.addOptions(
        {
            label: `${groupedQuery[i][0].number}.
${groupedQuery[i][0].name}`,
            value: `${i}`,
        },
    );
}
const selectLessonRow = new ActionRowBuilder().addComponents(selectLesson);
```

Фиг. 22: Програмен код на SelectMenu, използван за избор на урок

За да извършва дейност, падащият списък се нуждае от т.нар. **колектор** (collector). Колектор е вид слушател (listener), в който се помещава програмния код, който искаме да се изпълни при правене на избор от списъка. Колекторът може да се „ослушва“ за четири вида събития. В тази разработка са използвани само два от тях, тъй като другите не са необходими. Използваните събития са **collect** (събиране) и **end** (край), като **collect** възниква при направен избор от менюто, а **end** възниква при стопиране на колектора (поради изтекло време за слушане или ръчно спиране) (**Фиг. 23**).

```
const selectCollector = await
initialMessage.createMessageComponentCollector({
    componentType: ComponentType.StringSelect,
    time: 300000,
});
selectCollector.on("collect", async (i) => {
    switch (i.customId) {
        case 'selectLesson': { ... };
        case 'selectImage': { ... };
        default: { break; }
    }
});
selectCollector.on("end", async collected => { ... });
```

Фиг. 23: Програмен код на колектор, използван при реализацията на команда /уроци

С цел представяне на информацията в структуриран и подреден вид, всички съобщения на бота са оградени в ембед (с изключение на уведомителните такива). Ембед се образува с помощта на класа `EmbedBuilder()` на `discord.js`, който позволява и персонализирано конфигуриране. Възможно е задаване на текст на описание, заглавие, автор, цвят, линкове, полета и др. Пример за програмна реализация на ембед може да бъде видян на **Фиг. 24**.

```
const initialEmbed = new EmbedBuilder()
  .setDescription(
    `**Предмет:** ${subject}\n**Клас:** ${grade}` +
    `\n**Брой въпроси:** ${questionsAmount}` +
    `\n**Времетраене:** ${minutes} мин. и ${seconds} сек.` +
    `\n\nСъвет: Имай предвид, че ще можеш да избереш отговор само веднъж - избирай внимателно!`
  )
  .setTitle(`Тест № ${selectedTest + 1} ще стартира след ${testStartingTime / 1000} секунди...`)
  .setColor(`${process.env.embedColor}`);
```

Фиг. 24: Ембед, използван при реализацията на команда /тестове

2.1.5. Програмна реализация на валидация на разрешения

За да работи коректно, приложението `StudyBuddy` се нуждае от два вида сървърни разрешения в Дискорд: **SendMessages** (изпраща съобщения) и **ManageMessages** (управлява съобщения). Това налага извършването на проверка още в самото начало на всяко изпълнение на команда, която изисква обратна връзка от бота. Именно с тази цел е разработена функцията `hasPermissions()` (**Фиг. 25**).

```
async function hasPermissions(interaction) {
  if
    (!interaction.guild.members.me.permissions.has(PermissionsBitField.Flags.ManageMessages)) {
    interaction.reply('Услугите ми не са налични, тъй като не разполагам с необходимото разрешение за редактиране на съобщения в този канал. Собственикът на този канал бе уведомен за това.');
```

```
    const guildOwner = await interaction.guild.fetchOwner();
    const embed = new EmbedBuilder()
      ...
    guildOwner.send({ embeds: [embed] });
    return false;
  }
  else if
    (!interaction.guild.members.me.permissions.has(PermissionsBitField.Flags.SendMessages)) {
    interaction.member.send('Не разполагам с необходимите разрешения за изпращане и редактиране на съобщения в този канал.');
```

```
    const guildOwner = await interaction.guild.fetchOwner();
    const embed = new EmbedBuilder()
      ...
    guildOwner.send({ embeds: [embed] });
    return false;
  } return true; }
```

Фиг. 25: Програмен код на функция `hasPermissions()`

Ако ботът разполага с разрешение да изпраща съобщения, но не и да ги управлява, то той изпраща уведомително съобщение до потребителя в текстовия канал на сървъра, в който е била извикана командата. Ако ботът не разполага с разрешение да изпраща съобщения, то той изпраща уведомително съобщение до потребителя чрез функцията за директни съобщения на Дискорд (т.е. извън сървъра). И в двата случая се изпраща ембед до създателя на неконфигурирания сървър, чрез който се указва необходимата информация за проследяване и разрешаване на случая.

2.2. Реализация на уебсайт с наръчник за употреба

2.2.1. Структура

Главната директория на проекта на уебсайта съдържа следните файлове и папки:

- **index.html** – главен и единствен описващ файл на уебсайта;
- **styles.css** – файл за каскадни стилове, описващ стиловете и форматирането на различни html елементи от index.html;
- **scripts.js** – изпълним файл, съдържащ скриптове на необходимата логика за реализирането на интерактивни функции;
- **jquery-3.6.4.min.js** – файл на библиотеката jQuery;
- **images** – папка, съдържаща всички изображения, използвани в уебсайта (аватар, демонстративни изображения за ръководството).

2.2.2. Програмна реализация на уебсайта

Уебсайтът е проектиран да използва SPA модела.

SPA (Single Page Application) е модел за проектиране на уеб приложение, при който цялото приложение е поместено в една единствена уеб страница. Употребата на SPA позволява динамична актуализация на съдържанието на страница, осигурявайки по-плавно и по-интерактивно потребителско изживяване.

Страницата, която описва целия уебсайт, е **index.html**. В главата се помещават връзките с необх. файлове styles.css, scripts.js и jquery-3.6.4.min.js, а в тялото – самото съдържание на уебсайта. Структурата на тялото се състои от лента с навигация, заглавка и съвкупност от контейнери. Контейнерите съдържат двата изгледа: начален и ръководство.

Началният изглед има за цел да въведе посетителя в естеството на уебсайта. В него е поместено кратко описание на приложението StudyBuddy, включително списък с неговите функции. Дизайнът на изгледа е проектиран да наподобява интерфейса на Дискорд (тъмна тема).

Изглед „Ръководство“ предоставя на посетителите (и потенциални потребители на StudyBuddy) подробна информация за това как се работи с бота. Тук са разположени командите, като към всяка команда е включено обяснение стъпка по стъпка. За допълнително улеснение са поставени и изображения. Всяка команда е отделена в свой собствен контейнер, като всяка стъпка е описана с `` елемент.

Поради факта, че уебсайтът е SPA, за навигацията между различните изгледи бе необходимо да се установи логика. Тя е представена чрез скрипт, написан на JavaScript (с помощ от библиотеката jQuery) и е отделена във файл `scripts.js`. Скриптът се зарежда едновременно с уебсайта и се изпълнява при необходимост (**Фиг. 28**).

```
$(document).ready(function () { // Изпълнява се при зареждане на уебсайта

    $("nav.navigation").on("click", "a.parent", function (e) { // Изпълнява
    се при кликуване върху а елементи с клас parent от nav.navigation
        e.preventDefault(); // Предотвратява препратката на елемента а

        $("nav.navigation a.parent").removeClass("active"); // Премахва
        active класа от всички а елементи с клас parent
        $(this).addClass("active"); // Добавя класа active към кликнатия а
        елемент

        let spa = $(this).attr("spa"); // Променлива, съдържаща стойността
        на атрибута spa на кликнатия а елемент

        $("div.content > div").hide(); // Скрива всички подчинени див
        елементи на div.content
        $("div.content").find("div." + spa).show(); // Намира дива с класа,
        отговарящ на този, който потребителя иска да се визуализира и го показва

        if (spa == "commands") // При кликнат бутон 'Ръководство'
            $("nav.navigation div.commands").slideDown("fast"); //
            Визуализиране на шорткѐти към отделните команди
        else $("nav.navigation div.commands").slideUp("fast"); // Скриване
        на шорткѐтите

    });
});
```

Фиг. 28: Програмен код на `scripts.js`

Уебсайтът може да бъде достъпен на адрес <http://studybuddy.x10.mx/>.

3. Тестване на приложението StudyBuddy

3.1. Тестване на производителност (performance testing)

Selenium е рамка (framework) за тестване на софтуер, широко използвана за автоматизиране на уеб браузъри. Тя предоставя набор от инструменти и библиотеки, които позволяват на тестерите да автоматизират тестването на уеб приложения в различни браузъри и платформи.

Извършени са следните тестове за проверка бързодействието на приложението:

➤ **Замерване на времето за отговор на бота (команда /материали)**

Целта на теста е проверка на времето, необходимо на бота да визуализира заявен от потребител ресурс чрез команда „/материали география и икономика 5 изображение“. Целевото време за изпълнение е в рамките на 2 секунди. Извършени са 5 теста за по-точна преценка. **Таблица 2** показва постигнатото време на изпълнение на операцията.

Тест №	Време на изпълнение	Средно време на изпълнение
1	1,547 сек.	1,134 сек.
2	1,031 сек.	
3	1,032 сек.	
4	1,047 сек.	
5	1,015 сек.	

Таблица 2: Резултати от теста

Програмната реализация на теста може да бъде видяна на **Фиг. 29**.

```
[TestMethod]
public void MaterialsSpeedTest() {
    init();
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    driver.FindElement(By.XPath("/html/body/div[1]/div[2]/div[1]/div[1]/div/div[2]/div
/div/div/div/div[3]/div[2]/main/form/div/div[1]/div/div[3]/div/div/div/span/span/s
pan")).SendKeys("/");
    wait.Until(ExpectedConditions.ElementExists(By.Id("autocomplete-0"))).Click();
    // команда /материали
    driver.FindElement(By.Id("autocomplete-0")).Click(); // предмет: география
    driver.FindElement(By.Id("autocomplete-0")).Click(); // клас: 5
    driver.FindElement(By.Id("autocomplete-1")).Click(); // вид: изображение
    wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div
[1]/div[1]/div/div[2]/div/div/div/div/div[3]/div/main/form/div/div[2]/div/div[2]/d
iv/div/div/div/span[7]/span/span"))).SendKeys(Keys.Enter);
    wait.Until(ExpectedConditions.ElementExists(By.ClassName("select-
Zz0IcO"))).Click();

    wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div
[1]/div[3]/div/div[1]/div[1]/div"))).Click();
    long startTime = Environment.TickCount; // Начало на замерване
    wait.Until(ExpectedConditions.ElementExists(By.ClassName("grid-1aWVsE"))); //
    Паузира изпълнение, докато не се появи елемента
    long endTime = Environment.TickCount; // Край на замерване
    long totalTime = endTime - startTime; // Отнето време за отговор
    int targetTime = 2000; // Целево време за отговор (максимум)
    Console.WriteLine("Изпълнението отне " + (double)totalTime / 1000 + " сек.");
    Assert.IsTrue(totalTime <= targetTime); }
```

Фиг. 29: Програмен код на теста върху команда /материали

➤ **Замерване на времето за отговор на бота (команда /уроци)**

Целта на теста е проверка на времето, необходимо на бота да визуализира заявен от потребител ресурс чрез команда „/уроци география и икономика 5“. Целевото време за изпълнение е в рамките на 2 секунди. Извършени са 5 теста за по-точна преценка. **Таблица 3** показва постигнатото време на изпълнение на операцията.

Тест №	Време на изпълнение	Средно време на изпълнение
1	1,031 сек.	1,137 сек.
2	1,031 сек.	
3	1,047 сек.	
4	1,547 сек.	
5	1,031 сек.	

Таблица 3: Резултати от теста

Програмната реализация на теста може да бъде видяна на **Фиг. 30**.

```
[TestMethod]
public void LessonsSpeedTest ()
{
    init();
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

    driver.FindElement(By.XPath("/html/body/div[1]/div[2]/div[1]/div[1]/div/div[2]/div
/div/div/div/div[3]/div[2]/main/form/div/div[1]/div/div[3]/div/div/div/span/span/s
pan")).SendKeys("/");
    wait.Until(ExpectedConditions.ElementExists(By.Id("autocomplete-3"))).Click();
    // команда /уроци
    driver.FindElement(By.Id("autocomplete-0")).Click(); // предмет: география
    driver.FindElement(By.Id("autocomplete-0")).Click(); // клас: 5

    wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div
[1]/div[1]/div/div[2]/div/div/div/div[2]/div[3]/div/main/form/div/div[2]/div/div[2]
/div/div/div/span[5]/span/span"))).SendKeys(Keys.Enter);
    wait.Until(ExpectedConditions.ElementExists(By.ClassName("select-
Zz0IcO"))).Click();

    wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div
[1]/div[3]/div/div[1]/div[1]/div"))).Click();

    long startTime = Environment.TickCount;
    wait.Until(ExpectedConditions.ElementExists(By.ClassName("grid-1aWVsE")));
    long endTime = Environment.TickCount;

    long totalTime = endTime - startTime; // Отнето време за отговор
    int targetTime = 2000; // Целево време за отговор (максимум)

    Console.WriteLine("Изпълнението отне " + (double)totalTime / 1000 + " сек.");
    Assert.IsTrue(totalTime <= targetTime);
}
```

Фиг. 30: Програмен код на теста върху команда /уроци

- **Замерване на времето, което отнема бота при изпълнение на тест (команда /тестове)**

Целта на теста е проверка колко от времето за изпълнение на тест на потребителя отнема ботът. Целевото време за изпълнение е в рамките на 20 секунди. Извършени са 5 теста за по-точна преценка. **Таблица 4** показва отнетото от потребителя време за изпълнение на теста.

Тест №	Отнето време за изпълнение	Средно отнето време за изпълнение
1	17,079 сек.	17,212 сек.
2	17,625 сек.	
3	17,094 сек.	
4	17,094 сек.	
5	17,172 сек.	

Таблица 4: Резултати от теста

Програмната реализация на теста може да бъде видяна на **Фиг. 31**.

```
public void TestsSpeedTest()
{
    init();
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    driver.FindElement(By.XPath("/html/body/div[1]/div[2]/div[1]/div[1]/div/div[2]/div/div/div/div/div[3]/div[2]/main/form/div/div[1]/div/div[3]/div/div/div/span/span/span")) .SendKeys("/");
    wait.Until(ExpectedConditions.ElementExists(By.Id("autocomplete-2"))).Click();
    // команда /тестове
    driver.FindElement(By.Id("autocomplete-0")).Click(); // предмет: география
    driver.FindElement(By.Id("autocomplete-0")).Click(); // клас: 5
    wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div[1]/div[1]/div/div[2]/div/div/div/div[2]/div[3]/div/main/form/div/div[2]/div/div[2]/div/div/div/span[5]/span/span")) .SendKeys(Keys.Enter);
    wait.Until(ExpectedConditions.ElementExists(By.ClassName("select-Zz0IcO"))).Click();
    wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div[1]/div[3]/div/div[1]/div[1]/div"))).Click();

    long startTime = Environment.TickCount;
    wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div[1]/div[1]/div/div[2]/div/div/div/div[2]/div[3]/div[2]/main/div[1]/div/div/ol/li[1]/div/div[3]/div/div[2]/div/div/div"))).Click(); // Клик в/у меню отговор
    wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div[1]/div[3]/div/div[1]/div[1]"))).Click();

    for (int i = 2; i <= 7; i++)
    {
        Thread.Sleep(1000);
        wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div[1]/div[1]/div/div[2]/div/div/div/div[2]/div[3]/div[2]/main/div[1]/div/div/ol/li[1]/div/div[3]/div/div[1]/div/div/div"))).Click(); // Клик в/у меню въпрос

        wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div[1]/div[3]/div/div[1]/div[" + i + "]"))).Click(); // Клик върху следващ въпрос
        Thread.Sleep(1000);
        wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div[1]/div[1]/div/div[2]/div/div/div/div[2]/div[3]/div[2]/main/div[1]/div/div/ol/li[1]/div/div[3]/div/div[2]/div/div/div"))).Click(); // Клик в/у меню отговор
        wait.Until(ExpectedConditions.ElementExists(By.XPath("/html/body/div[1]/div[2]/div[1]/div[3]/div/div[1]/div[1]"))).Click(); // Клик върху отговор а)
    }

    long endTime = Environment.TickCount;
    long totalTime = endTime - startTime; // Отнето време за отговор
    int targetTime = 20000; // Целево време за отговор (максимум)
    Console.WriteLine("Изпълнението отне " + (double)totalTime / 1000 + " сек.");
    Assert.IsTrue(totalTime <= targetTime);
}
```

Фиг. 31: Програмен код на теста върху команда /тестове

Глава 4. Ръководство за работа с приложението

1. Инсталация и стартиране

1.1. Инсталация

За да бъде стартирано, приложението StudyBuddy се нуждае от следния софтуер:

- **Node.js (v16.9.0+)** с инсталирани модули „discord.js“ (v14.8.0), „dotenv“ (v16.0.3), „mysql“ (v2.18.1), „@discordjs/voice“ (v0.16.0), „@discordjs/opus“ (v0.9.0), „ffmpeg-static“ (v5.1.0) и „sodium“ (v3.0.2);
- **XAMPP** с инсталирани модули Apache и MySQL.

След инсталацията на необходимия софтуер е нужно да бъде добавена базата данни на StudyBuddy. За целта се стартират услугите Apache и MySQL от контролния панел XAMPP, достъпва се phpMyAdmin (<http://localhost/phpmyadmin/>) и през него се импортира .sql файла на проекта.

1.2. Стартиране

За да можем да взаимодействаме с бота, разбира се първо трябва да го добавим в нашия сървър в Дискорд. Това може да бъде направено през уебсайта на StudyBuddy (<http://studybuddy.x10.mx/>).

Стартирането на приложението става чрез изпълнение на команда „**node index.js**“ в отворен терминал (напр. command prompt) в главната директория на проекта. Когато ботът е готов за употреба, в терминала ще бъде изведено съобщение „Ботът бе стартиран успешно.“.

2. Ръководство за използване

2.1. Работа с команда /материали

Командата се използва за достъп до специфичен материал. За да се използва е необходимо с изпращането ѝ да се укажат предмет, клас и вид материал.

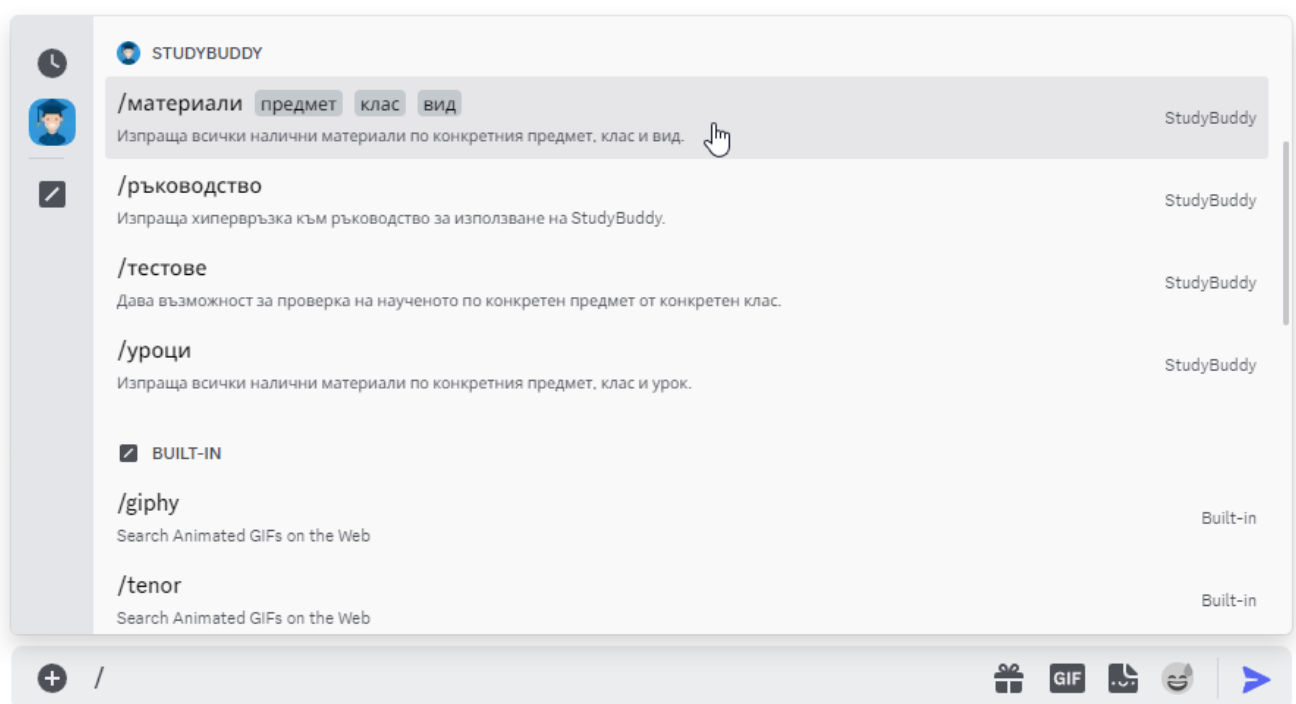
Стъпки за употреба:

- 1) Въведете знак '/' (без скобите) в текстов канал и от появилия се прозорец изберете командата /материали (**Фиг. 32.1**);
- 2) Изберете предмет, клас и вид материал (**Фиг. 32.2**);
- 3) Изпратете командата в канала;
- 4) Изберете урок от появилото се меню (**Фиг. 32.3**);

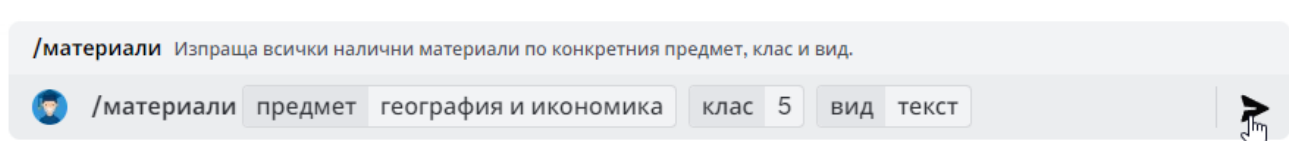
Ако за вид материал е избран текст, изображение или видеоклип, след стъпка 4 приложението ще визуализира заявената информация.

Стъпките за възпроизвеждане на аудиоклип при избран вид аудиоклип са следните:

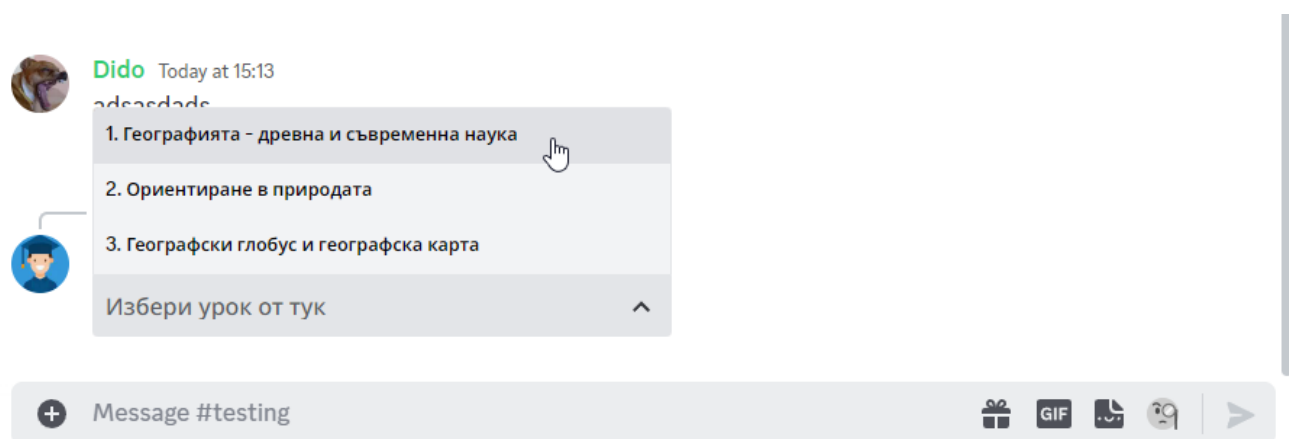
- 1) Присъединете се към гласов канал в сървъра;
- 2) Изберете аудиоклип за възпроизвеждане от менюто.



Фиг. 32.1: Интерфейс на Дискорд, показващ списък с налични команди



Фиг. 32.2: Попълнена команда /материали



Фиг. 32.3: Избор на урок от меню

2.2. Работа с команда /уроци

Командата се използва за структурирано извеждане на уроци. За да се използва е необходимо с изпращането ѝ да се укажат предмет и клас.

Стъпки за употреба:

- 1) Въведете знак '/' (без скобите) в текстов канал и от появилия се прозорец изберете командата /уроци (по същия начин както е показано на **Фиг. 32.1**);
- 2) Изберете предмет и клас (по същия начин както е показано на **Фиг. 32.2**);
- 3) Изпратете командата в канала;
- 4) Изберете урок от появилото се меню (**Фиг. 32.3**).

След стъпка 4 приложението визуализира заявената информация. Потребителят може да навигира между различните уроци и/или изображения по всяко време, използвайки съответните менюта. Ако някое меню се скрие, потребителят ще трябва да изпрати командата отново.

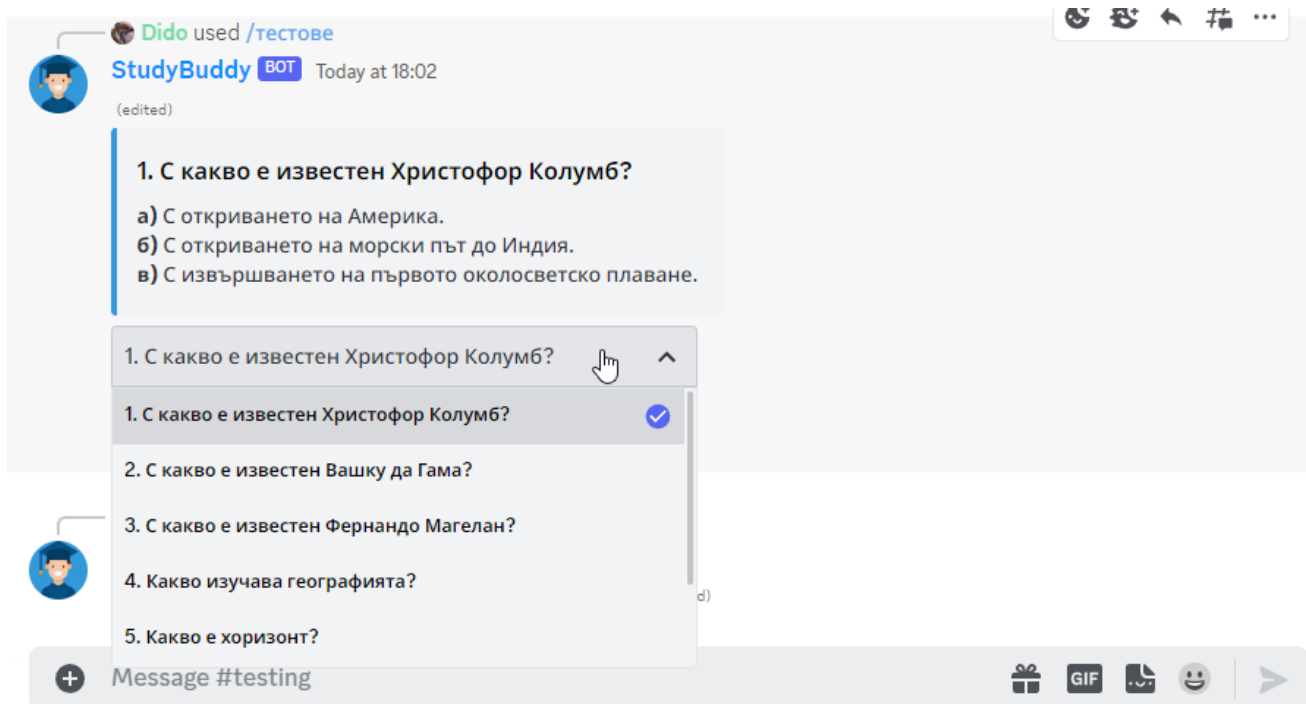
2.3. Работа с команда /тестове

Командата се използва за извършване на самооценка на знания чрез тестове. За да се използва е необходимо с изпращането ѝ да се укажат предмет и клас.

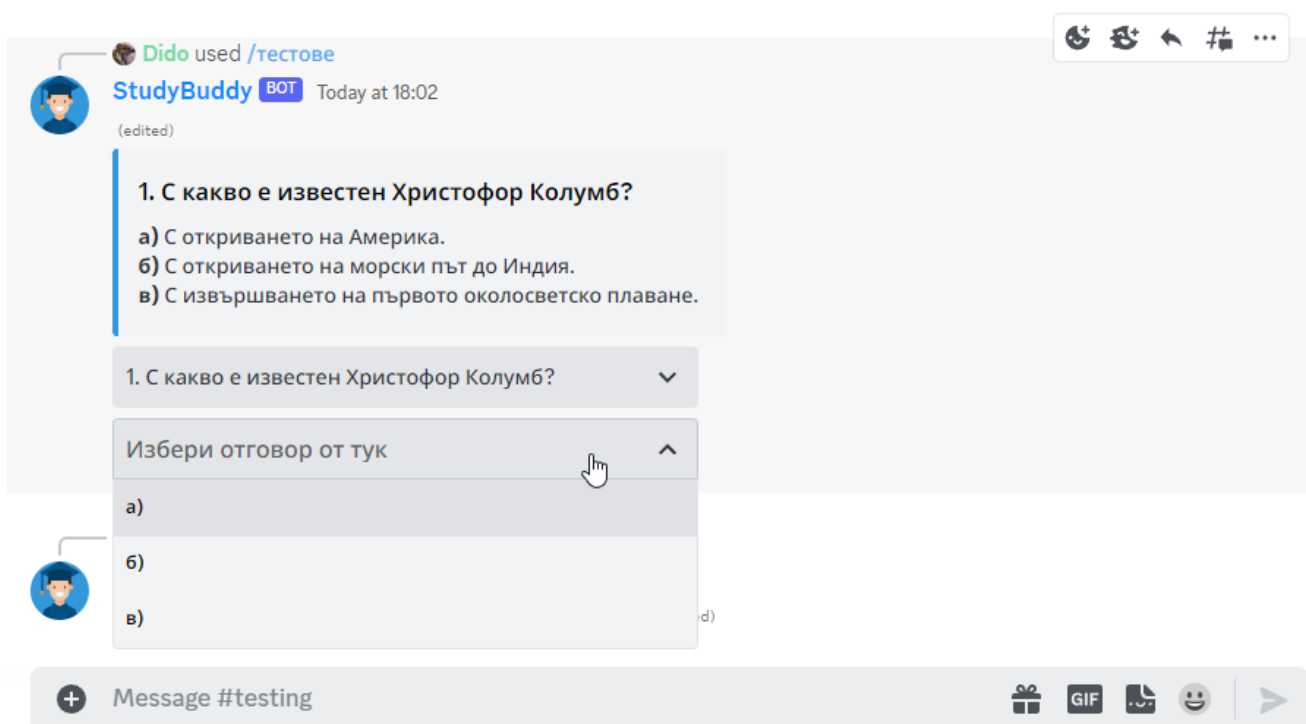
Стъпки за употреба:

- 1) Въведете знак '/' (без скобите) в текстов канал и от появилия се прозорец изберете командата /тестове (по същия начин както е показано на **Фиг. 32.1**);
- 2) Изберете предмет и клас (по същия начин както е показано на **Фиг. 32.2**);
- 3) Изпратете командата в канала;
- 4) Изберете тест от появилото се меню (по същия начин както е показано на **Фиг. 32.3**);
- 5) Прочетете внимателно съобщението, което StudyBuddy изпрати. То съдържа ценна информация, като брой въпроси в теста и време за изпълнение. Разполагате с 30 секунди, след които ботът ще покаже първия въпрос;
- 6) Започнете да попълвате теста. Можете да навигирате между различните въпроси от теста по всяко време, използвайки менюто за избор на въпрос (**Фиг. 33.1**). За избор на отговор се използва менюто за избор на отговор (**Фиг. 33.2**).

След като отговорите на всички въпроси, ботът ще покаже резултатите от теста (**Фиг. 33.3**). Ако не сте успели да отговорите на всички въпроси в рамките на времето за изпълнение, ботът ще покаже резултатите въз основа на отговорените до момента.



Фиг. 33.1: Избор на въпрос от меню



Фиг. 33.2: Избор на отговор от меню

Предмет: География и икономика Клас: 5

ТЕСТ № 1 — РЕЗУЛТАТИ

Поздравления, DilyanD! Ти изкара отлична оценка!

Твоята оценка е: Отличен 6

Успя да отговориш правилно на 7 от 7 отговорени въпроса от общо 7.

Изпълнението на теста ти отне 1 минути и 24 секунди.

1. С какво е известен Христофор Колумб?



Посочен отговор: а) С откриването на Америка.

Правилен отговор: а) С откриването на Америка.

2. С какво е известен Вашку да Гама?



Посочен отговор: б) С откриването на морски път до Индия.

Правилен отговор: б) С откриването на морски път до Индия.

3. С какво е известен Фернандо Магелан?



Посочен отговор: в) С извършването на първото околосветско плаване.

Правилен отговор: в) С извършването на първото околосветско плаване.

4. Какво изучава географията?



Посочен отговор: а) Природни обекти (планини, реки, гори и др.) и обекти, създадени от човека (страни, градове, пътища и др.)

Правилен отговор: а) Природни обекти (планини, реки, гори и др.) и обекти, създадени от човека (страни, градове, пътища и др.)

5. Какво е хоризонт?



Фиг. 33.3: Визуализация на резултат от тест при изкарана отлична оценка

Глава 5. Изводи и препоръки

Приложението StudyBuddy е едно съвременно решение. Разработването и внедряването му демонстрира неговия потенциал като ефективен образователен инструмент за ученици в прогимназиалния етап. Възползвайки се от функционалностите на Дискорд, ботът създава виртуална учебна среда, която насърчава ангажираността и сътрудничеството между младежите и ги подтиква към самообучение.

Въпреки че на този етап ботът изглежда обещаващо, необходими са допълнителни проучвания и оценяване, за да се открие неговата дългосрочна ефективност, мащабируемост и адаптивност към различни образователни среди.

В бъдеще може да бъдат въведени следните промени и подобрения:

- Промяна в дизайна и структурата на приложението, така че предметите и класовете също да се определят динамично в зависимост от данните в базата данни;
- Добавяне на възможност за манипулация на базата данни директно през Дискорд;
- Добавяне на нови учебни материали.

III. Заключение

Въз основа на информацията, представена в тази дипломна записка, може да се направи извода, че възложените цели в дипломния проект са изпълнени. Предложеното решение предоставя всичко необходимо:

- База данни, съдържаща учебни материали по география и икономика, история и цивилизации и информационни технологии;
- Разработено приложение с привлекателен и интерактивен интерфейс със следните възможности:
 - Възможност за възпроизвеждане и визуализация на мултимедия – текст, изображения, видеоклипове и аудиоклипове;
 - Възможност за самооценяване върху направление по избор.
- Уебсайт, съдържащ наръчник за употреба, представен на достъпен и разбираем език за подрастващите.

Разработеният Дискорд бот StudyBuddy е резултат от обединение между Дискорд бот за общо приложение и образователен чатбот.

IV. Литературни източници (конвенция APA)

- [6] Редактори на Уикипедия (2022). Игровизация. Уикипедия, Свободната енциклопедия. Достъп на 17.03.2023 г. от <https://bg.wikipedia.org/wiki/Игровизация>
- [14] Сътрудници на MDN (2023). JavaScript. MDN Web Docs. Достъп на 08.05.2023 г. от <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [16] Сътрудници на GitHub (2023). About npm. Npm Docs. Достъп на 08.05.2023 г. от <https://docs.npmjs.com/about-npm>
- [17] Създатели на Discord.js (2023). Documentation. Discord.js. Достъп на 08.05.2023 г. от <https://old.discordjs.dev/#/docs/discord.js/main/general/welcome>
- [20] Сътрудници на GitHub (2023). Mysql Documentation. Npmjs. Достъп на 09.05.2023 г. от <https://www.npmjs.com/package/mysql?activeTab=readme>
- [21] Сътрудници на phpMyAdmin (2023). About. phpMyAdmin. Достъп на 09.05.2023 г. от <https://www.phpmyadmin.net/>
- [24] Сътрудници на MDN (2023). What is CSS?. MDN Web Docs. Достъп на 09.05.2023 г. от https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS
- [25] Сътрудници на jQuery (2015). About jQuery. jQuery Learning Center. Достъп на 09.05.2023 г. от <https://learn.jquery.com/about-jquery/>
- [26] Сътрудници на VS Code (2023). VS Code Documentation. Visual Studio Code. Достъп на 09.05.2023 г. от <https://code.visualstudio.com/docs>
- [9] Ari Horesh (2023). StudyLion. Top.gg. Достъп на 19.03.2023 г. от <https://top.gg/bot/889078613817831495>
- [22] Ashwani K. (2021). What is XAMPP and how to install XAMPP?. DevOpsSchool. Достъп на 09.05.2023 г. от <https://www.devopsschool.com/blog/what-is-xampp-and-how-to-install-xampp/>
- [23] Astari S. (2023). What is HTML? Hypertext Markup Language Basics Explained. Hostinger Tutorials. Достъп на 09.05.2023 г. от <https://www.hostinger.com/tutorials/what-is-html>
- [11] Countryvia (2023). Countryvia. Top.gg. Достъп на 19.03.2023 г. от <https://top.gg/bot/859024110905458718>
- [19] Chad Brooks (2023). What is SQL?. BusinessNewsDaily. Достъп на 09.05.2023 г. от <https://www.businessnewsdaily.com/5804-what-is-sql.html>

- [1] Duolingo (2023). Approach. Duolingo. Достъп на 10.03.2023 г. от <https://www.duolingo.com/approach>
- [13] Dev account (2023). Geocord. Top.gg. Достъп на 21.03.2023 г. от <https://top.gg/bot/1014206103493558342>;
- [27] Discord.js. (2022). Embed Preview (Screenshot). Discord.js Guide. Достъп на 15.05.2023 г. от <https://discordjs.guide/popular-topics/embeds.html#embed-preview>
- [8] Joakim (2023). Jockie Music. Top.gg. Достъп на 18.03.2023 г. от <https://top.gg/bot/411916947773587456>
- [3] Kahoot DragonBox (2022). About Us. DragonBox. Достъп на 10.03.2023 г. от <https://dragonbox.com/about>
- [4] Kahoot DragonBox (2022). Products. DragonBox. Достъп на 10.03.2023 г. от <https://dragonbox.com/products>
- [7] MEE6 (2023). MEE6. Top.gg. Достъп на 18.03.2023 г. от <https://top.gg/bot/159985870458322944>
- [12] Nooticulus (2023). Archimedes. Top.gg. Достъп на 21.03.2023 г. от <https://top.gg/bot/863993305008439306>
- [18] Oracle (2023). What is MySQL?. MySQL. Достъп на 09.05.2023 г. от <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- [5] Quynh Nguyen (2022). No-stress English Learning! 5 Chat Robots That'll Never Judge You. FluentU. Достъп на 11.03.2023 г. от <https://www.fluentu.com/blog/english/chat-with-robot-to-learn-english/>
- [10] Team TriviaBot (2023). TriviaBot. Top.gg. Достъп на 19.03.2023 г. от <https://top.gg/bot/715906723982082139>
- [15] Taha Sufiyan (2023). What is Node.js: A Comprehensive Guide. SimpliLearn. Достъп на 08.05.2023 г. от <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs>
- [2] Vikash Gupta (2023). Duolingo Review - Can It Help You Speak A Language?. StudyFrenchSpanish. Достъп на 10.03.2023 г. от <https://www.studyfrenchspanish.com/duolingo-review/>