Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**КУРСОВАЯ РАБОТА**

По дисциплине «Математические основы интеллектуальных систем»

Тема: «Проектирование и реализация веб-приложения сборки кубика Рубика»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Козинский А.А.

Брест 2023

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ**3

**1. СИСТЕМНЫЙ АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ**6

1.1 Основные положения теории групп перестановок (на примере венгерского кубика Рубика)6

1.2 Обзор и анализ существующих приложений для сборки кубика7

1.3 Выбор средств для создания веб-приложения8

1.4 Постановка задачи на создание веб-приложения10

**2. РЕЗУЛЬТАТЫ СОЗДАНИЯ ВЕБ-ПРИЛОЖЕНИЯ**13

2.1 Структура веб-приложения сборки кубика Рубика13

2.2 Структура информационного обеспечения системы14

2.3 Структура пользовательского интерфейса приложения14

**3. РЕЗУЛЬТАТЫ И ИСПЫТАНИЯ ВЕБ-ПРИЛОЖЕНИЯ**16

3.1 Реализация веб-приложения сборки кубика Рубика16

3.2 Результаты испытаний17

**ЗАКЛЮЧЕНИЕ**21

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**23

**ПРИЛОЖЕНИЕ А.** ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ЗАПУСКЕ СЕРВЕРА26

**ПРИЛОЖЕНИЕ Б.** КОД ПРОЕКТА27

**ВВЕДЕНИЕ**

В настоящее время большинство людей в мире знакомо с головоломкой-кубиком Рубика, который был изобретен в 1974 году и до сих пор покоряет сердца миллионов по всему миру [4].

**Актуальность выбранной темы**

Разработка веб-приложения для сборки кубика Рубика является актуальной по следующим причинам:

1. Игра в кубик Рубика представляет собой захватывающее и умственно развивающее занятие, вокруг которого организовываются соревнования, направленные на достижение максимальной скорости сборки [3].
2. Математика кубика Рубика — совокупность математических методов для изучения свойств кубика Рубика с абстрактно-математической точки зрения [7].
3. Веб-приложения обладают рядом преимуществ, таких как кросс-платформенность, быстроту разработки и возможность размещения на облачных серверах [2]. Основной язык для разработки веб-приложений — JavaScript [17], который находится на 1-ом месте по популярности [10].
4. Распространенность интернета и доступность для пользователей со всего мира делают веб-приложение кубика Рубика актуальным, так как любой пользователь с доступом к интернету может играть и наслаждаться этой головоломкой, не зависимо от своего местоположения [2].

**Объектом исследования** является технологии реализации веб-приложений, которые позволяют пользователям собирать кубик Рубика виртуально с помощью интерфейса, доступного через веб-браузер.

**Предметом** является средства технологий, которые используются для реализации графических веб-приложений.

**Целью** данной курсовой работы является разработать веб-приложения для сборки кубика Рубика. Для реализации поставленной цели, необходимо решить следующие задачи:

1. Выполнить обзор теоретических положений, используемых для описания кубика Рубика.
2. Выполнить постановку задачи.
3. Разработать архитектуру веб-приложения для сборки кубика Рубика.
4. Реализовать компоненты архитектуры веб-приложения с использованием современных веб-технологий.
5. Провести тестирование приложения для обеспечения его работы.

Методы, используемые в работе, могут включать обзор существующих решений, проектирование интерфейса, программирование веб-приложения с использованием соответствующих технологий (например, HTML, CSS, JavaScript) и оценку эффективности разработанного приложения на основе тестирования и обратной связи пользователей.

В результате работы ожидается создание функционального веб-приложения для сборки кубика Рубика и демонстрация его работы и возможностей через прототип или рабочую версию приложения.

**Методы исследования:**

обзор существующих приложений, программных решений для кубика,

проектирование и разработка, тестирование и оценка, сравнительный анализ.

Курсовая работа предоставит возможность разработать полноценное веб-приложение для сборки кубика Рубика, а также позволит ознакомиться с принципами разработки программного обеспечения и применением современных веб-технологий.

**1. СИСТЕМНЫЙ АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ**

1.1. Основные положения теории групп перестановок (на примере венгерского кубика Рубика).

Теория групп перестановок является одной из основных областей алгебры, и она находит свое применение во многих областях науки и техники, включая криптографию, компьютерную графику, теорию кодирования и многие другие. В контексте венгерского кубика Рубика, группа перестановок состоит из всех возможных комбинаций поворотов граней кубика. Каждый поворот представляется как перестановка элементов кубика, а группа перестановок содержит все такие перестановки, которые могут быть получены путем комбинации различных поворотов граней кубика [7]. Существует 6 основных граней кубика, и каждая из них может быть повернута в одном из трех возможных направлений (по часовой стрелке на 90 градусов, против часовой стрелки на 90 градусов или на 180 градусов). Таким образом, существует 18 различных поворотов граней кубика, которые могут быть применены в любой последовательности. Группа перестановок венгерского кубика Рубика содержит 43 252 003 274 489 856 000 различных перестановок [19], что делает ее одной из самых больших групп перестановок, изучаемых в математике.

Основные положения теории групп перестановок, которые могут быть применены к венгерскому кубику Рубика, включают теорему Лагранжа, которая утверждает, что порядок любой подгруппы группы перестановок должен делиться на порядок группы [13]; теорему Кэли, которая говорит, что любую группу можно представить в виде группы перестановок [12]; и теорию конечных групп, которая изучает свойства групп с конечным числом элементов [14].

Более конкретно, теория групп перестановок может использоваться для изучения различных аспектов венгерского кубика Рубика, включая его различные комбинации, свойства и алгоритмы решения. Это делает ее важным инструментом для любого, кто хочет более глубоко понимать венгерский кубик Рубика и его механику.

1.2. Обзор и анализ существующих приложений для сборки кубика.

1. "Rubik's Cube Solver" (iOS, Android):

Это одно из самых популярных приложений для решения кубика Рубика. Оно предлагает различные методы сборки, включая начинающие и продвинутые стратегии. Приложение обеспечивает интуитивный интерфейс, позволяющий пользователю вводить текущую конфигурацию кубика и получать последовательность шагов для его сборки.

2. "Cube Timer" (iOS, Android):

Это приложение предназначено для отслеживания времени, затраченного на сборку кубика Рубика. Оно предлагает таймер, который позволяет фиксировать время начала и завершения сборки, а также подсчитывает среднее время сборки на основе предыдущих результатов. Приложение также предоставляет статистику и графики для анализа прогресса в сборке кубика.

3. "CFOP Trainer" (iOS):

Это приложение ориентировано на тренировку метода CFOP (Cross, F2L, OLL, PLL), который является одним из самых популярных методов сборки кубика Рубика. Оно предлагает различные сценарии для тренировки каждого этапа метода, помогая улучшить скорость и эффективность сборки.

4. "Rubik's Cube Algorithms" (iOS, Android):

Это приложение предоставляет обширную коллекцию алгоритмов для сборки кубика Рубика. Оно включает в себя алгоритмы для различных этапов CFOP, OLL (Orientation of the Last Layer) и PLL (Permutation of the Last Layer), а также другие полезные алгоритмы для ускорения сборки. Пользователи могут изучать алгоритмы и добавлять их в собственную коллекцию.

5. "Rubik's Cube Simulator" (Web):

Это веб-приложение, которое позволяет пользователям симулировать сборку кубика Рубика прямо на экране. Оно обеспечивает 3D-модель кубика и интерактивный интерфейс для вращения граней. Это позволяет пользователям экспериментировать с различными алгоритмами и стратегиями сборки, не трогая физический кубик. Такое приложение может быть полезным для новичков, которые только начинают изучать сборку кубика, а также для опытных пользователей, чтобы проверить различные подходы и улучшить свои навыки.

1.3. Выбор средств для создания веб-приложения

Среда разработки: PyCharm.

Языки программирования: Python, JavaScript, HTML, CSS.

Используемые библиотеки: Django, THREE.js, tween.js, lodash.js, Timer.js.

Django используется для разработки серверной части проекта, обеспечивая мощные инструменты для управления запросами, маршрутизации и базой данных. Он позволяет разрабатывать стабильные и масштабируемые веб-приложения с удобной архитектурой и функциональностью. В нем «из коробки» поддерживается работа с пользователями, встроена ORM для управления базой данных, шаблонизатор, а также много другого [8].

THREE.js является ключевым фреймворком для работы с 3D-графикой. Он предоставляет широкий набор функций для создания и отображения трехмерных объектов, управления камерами и освещением. С помощью THREE.js можно создавать впечатляющие визуализации кубика Рубика и реализовывать интерактивность для пользователя [1].

Tween.js предоставляет средства для создания плавных анимации и интерполяции значений. Он особенно полезен для создания анимированных переходов и эффектов при взаимодействии с кубиком Рубика [5].

Lodash.js является полезной утилитарной библиотекой, предоставляющей множество функций для обработки данных. Ее функции по работе с массивами, объектами и строками значительно упрощают манипуляции с данными и обеспечивают оптимизированную производительность кода [6].

Timer.js предоставляет возможности работы с таймерами и отслеживания времени. Он полезен для реализации функциональности, связанной с временными операциями, такими как отслеживание времени сборки кубика Рубика или установка таймеров для выполнения определенных действий.

1.4. Постановка задачи на создание приложения сборки кубика Рубика

Цель проекта:

Разработать веб-приложение, которое позволяет пользователям взаимодействовать с кубиком Рубика, сохранять время сборки в базе данных, просматривать список лидеров, а также регистрироваться и входить в свои аккаунты.

Основные функциональные требования:

• Интерактивное взаимодействие с кубиком Рубика: Приложение должно предоставлять возможность пользователю вращать грани кубика Рубика, чтобы проводить операции по сборке.

• Сохранение времени сборки: Приложение должно позволять пользователю сохранять время сборки кубика Рубика после успешного завершения сборки. Эти данные должны быть сохранены в базе данных, привязанные к каждому отдельному пользователю.

• Список лидеров: Приложение должно отображать список пользователей с наилучшими результатами сборки кубика Рубика. Список лидеров должен быть отсортирован по времени сборки, чтобы пользователи могли сравнивать свои результаты с другими.

• Регистрация и вход в аккаунт: Приложение должно предоставлять функциональность регистрации новых пользователей и возможность входа в свои аккаунты для уже зарегистрированных пользователей. Это обеспечит персонализированный опыт использования приложения и сохранение результатов сборки для каждого пользователя.

Дополнительные функциональные требования:

• Аутентификация и авторизация: Приложение должно обеспечивать безопасность данных путем реализации аутентификации и авторизации пользователей. Это позволит каждому пользователю иметь доступ только к своим данным и результатам сборки.

• Улучшенный дизайн и визуализация: Приложение должно иметь привлекательный и интуитивно понятный пользовательский интерфейс, который облегчит взаимодействие с кубиком Рубика и отображение списка лидеров.

Технические требования:

• Использование Django: Разработка приложения должна быть основана на фреймворке Django для обработки серверной части и взаимодействия с базой данных.

• Использование базы данных: Приложение должно использовать базу данных для хранения информации о пользователях, времени сборки и других необходимых данных.

• Фронтенд-разработка с использованием HTML, CSS и JavaScript: Для создания пользовательского интерфейса и взаимодействия с кубиком Рубика необходимо использовать HTML для разметки страниц, CSS для стилизации и JavaScript для обработки событий и динамического обновления данных.

• Использование AJAX: Для обмена данными между клиентской и серверной частями приложения рекомендуется использовать технологию AJAX, которая позволяет асинхронно отправлять и получать данные без перезагрузки страницы.

• Реализация аутентификации и авторизации: Для обеспечения безопасности данных и функциональности входа в аккаунт и регистрации пользователей необходимо реализовать механизм аутентификации и авторизации. Можно использовать встроенные средства аутентификации Django.

**2. РЕЗУЛЬТАТЫ СОЗДАНИЯ ВЕБ-ПРИЛОЖЕНИЯ**

2.1. Структура веб-приложения сборки кубика Рубика

Серверная часть приложения написана на Django. Django выполняет функцию маршрутизации (определяет маршруты URL и их соответствующие обработчики). Также Django предоставляет свою ORM для взаимодействия с базой данных.

Структура проекта Django [11]:

• manage.py: Файл, используемый для управления командами и настройками проекта Django.

• settings.py: Файл настроек Django, где определяются конфигурационные параметры, такие как база данных, аутентификация и статические файлы.

• urls.py: Файл, который определяет маршруты URL и их соответствующие обработчики (views) в Django.

• views.py: Файл, где определяются функции-обработчики запросов от клиента, включая обработку регистрации, входа в аккаунт, сохранения времени сборки и других операций.

• models.py: Файл, где определяются модели данных, такие как модель пользователя и модель времени сборки кубика Рубика.

• templates/: Директория, где размещаются шаблоны HTML, используемые для отображения пользовательского интерфейса.

• static/: Директория, где хранятся статические файлы, такие как CSS, JavaScript, изображения и другие ресурсы, необходимые для клиентской части приложения.

2.2. Структура информационного обеспечения системы

При отправке запроса на сервер, отработает соответствующий этому URL обработчик [16]. Например:

• При отправке запроса по URL “/” вернется HTML-код главной страницы.

• При отправке запроса по URL “/leaders/” обработчик получит отсортированные данные из базы данных, отрендерит HTML страницу, в зависимости от данных из базы данных, и вернет ее.

Django предоставляет стандартную модель пользователя, которая включает в себя имя пользователя, пароль, электронную почту. Для сохранения лучшего времени сборки кубика нужно добавить еще одно поле в модели, которое будет хранить количество миллисекунд [9]. При отправке нового времени сборки от клиента, в обработчике произойдет проверка, является ли новое время меньшим, чем текущее время. Если новое время меньше, чем текущее, то оно будет записано в базу данных.

2.3. Структура пользовательского интерфейса приложения

Главная страница состоит из сцены Three.js, на которой вырисовывается кубик, кнопок для перехода на страницу лидеров, входа в аккаунт, кнопок управления таймеров, кнопками управления кубиком Рубика.

Кубик можно вращать левой кнопкой мыши, приближать с помощью колесика. Можно вращать грани с помощью кнопок на экране или с помощью кнопок на клавиатуре (первая буква каждого цвета на английском языке кубика соответствует отвечающей за эту грань кнопке). Например, при нажатии кнопки “W” грань, в которой по центру находится белый цвет, будет повернута на 90° по часовой стрелке. Чтобы повернуть против часовой стрелки, нужно нажать сочетание клавиш “Shift + W”.

Для входа в аккаунт, нужно нажать кнопку “Войти”.

При запуске таймера, кубик сначала перемешается, затем обнулится, затем запуститься. Когда кубик будет собран, время отправится на сервер для сохранения его в базу данных.

1. **РЕАЛИЗАЦИЯ И ИСПЫТАНИЯ ВЕБ-ПРИЛОЖЕНИЯ**
   1. Реализация веб-приложения сборки кубика Рубика

Серверная часть приложения была реализована с использованием фреймворка Django. В качестве базы данных была выбрана sqlite3.

Были задействованы следующие модули:

* Модуль авторизации django.contrib.auth (предоставляет функциональность для аутентификации пользователей, управления учетными записями и обеспечения безопасности. С его помощью можно реализовать систему регистрации, входа в систему и управления правами доступа пользователей.)
* Модуль обработки форм django.forms (упрощает создание и обработку веб-форм)
* Модуль маршрутизации URL django.urls (позволяет определить маршруты URL для обработки запросов и направления их к соответствующим функциям или представлениям в приложении)
* Модуль базы данных django.db (предоставляет интеграцию с различными СУБД)
* Модуль представлений django.views (позволяет определить представления, которые обрабатывают запросы и возвращают соответствующие HTTP-ответы)

JavaScript, HTML и CSS были использованы для создания клиента веб-приложения. Были использованы библиотеки Three.js, tween.js, lodash.js, Timer.js, OrbitControl.js.

Были созданы:

* 3-д модель кубика Рубика (средствами Three.js).
* Кнопки и обработчики клавиш на клавиатуре (для поворота соответствующих граней).
* Кнопки для управления таймером, а также сам таймер с выводом на экран часов, минут, секунд, и сотен миллисекунд.
* Таблица лидеров в виде таблицы с выводом места пользователя, его именем и лучшем временем.
* Страница входа и регистрации аккаунтов.

3.2 Результаты испытаний

Для тестирования приложения необходимо установить и запустить проект. Порядок установки указан в Приложении А.

Открываем браузер и вводим в адресную строку 127.0.0.1:8000. Нас встречает главная страница приложения (рис. 1).

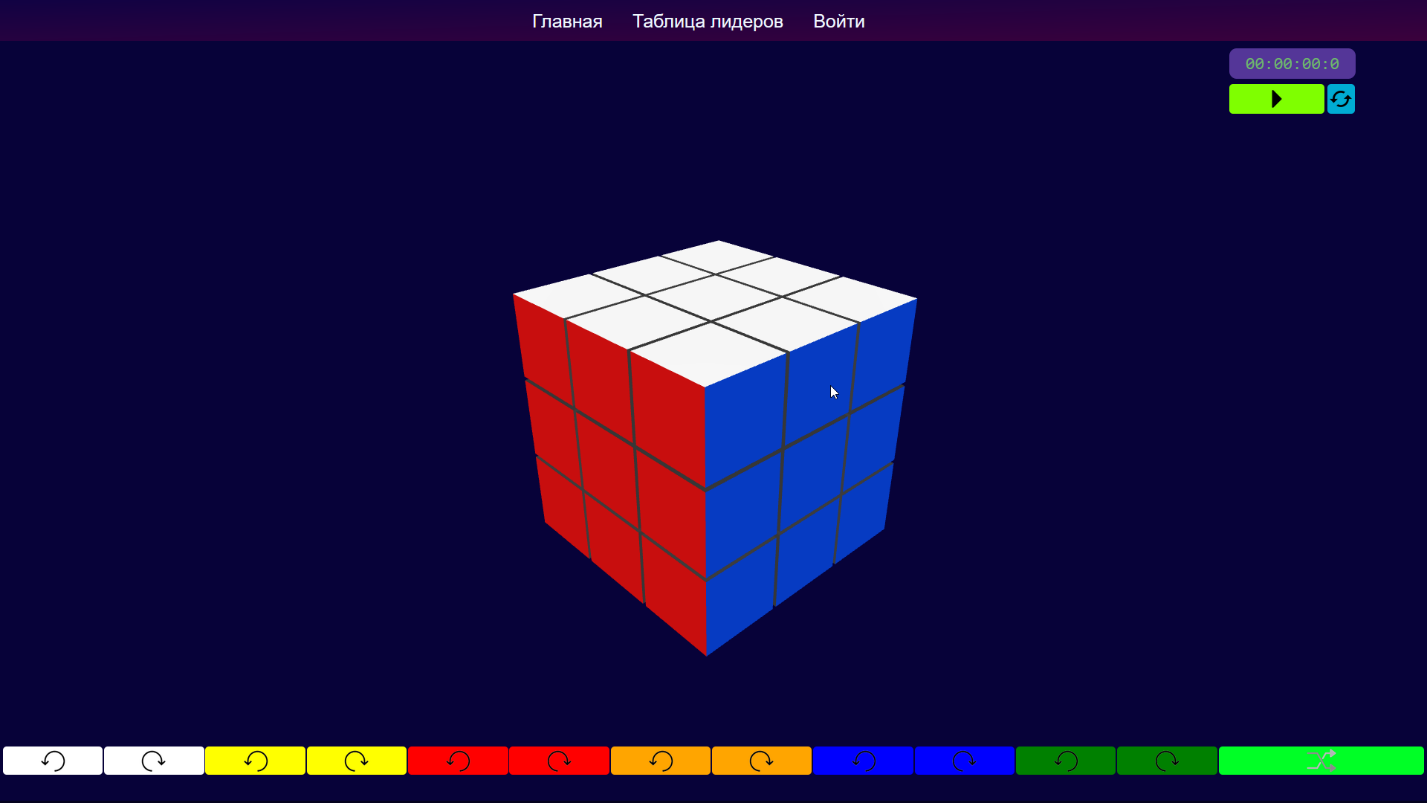


Рис. 1. Вид главной страницы

При нажатии кнопок управления кубиком, поворачиваются соответствующие грани, происходит перемешивание (рис. 2).

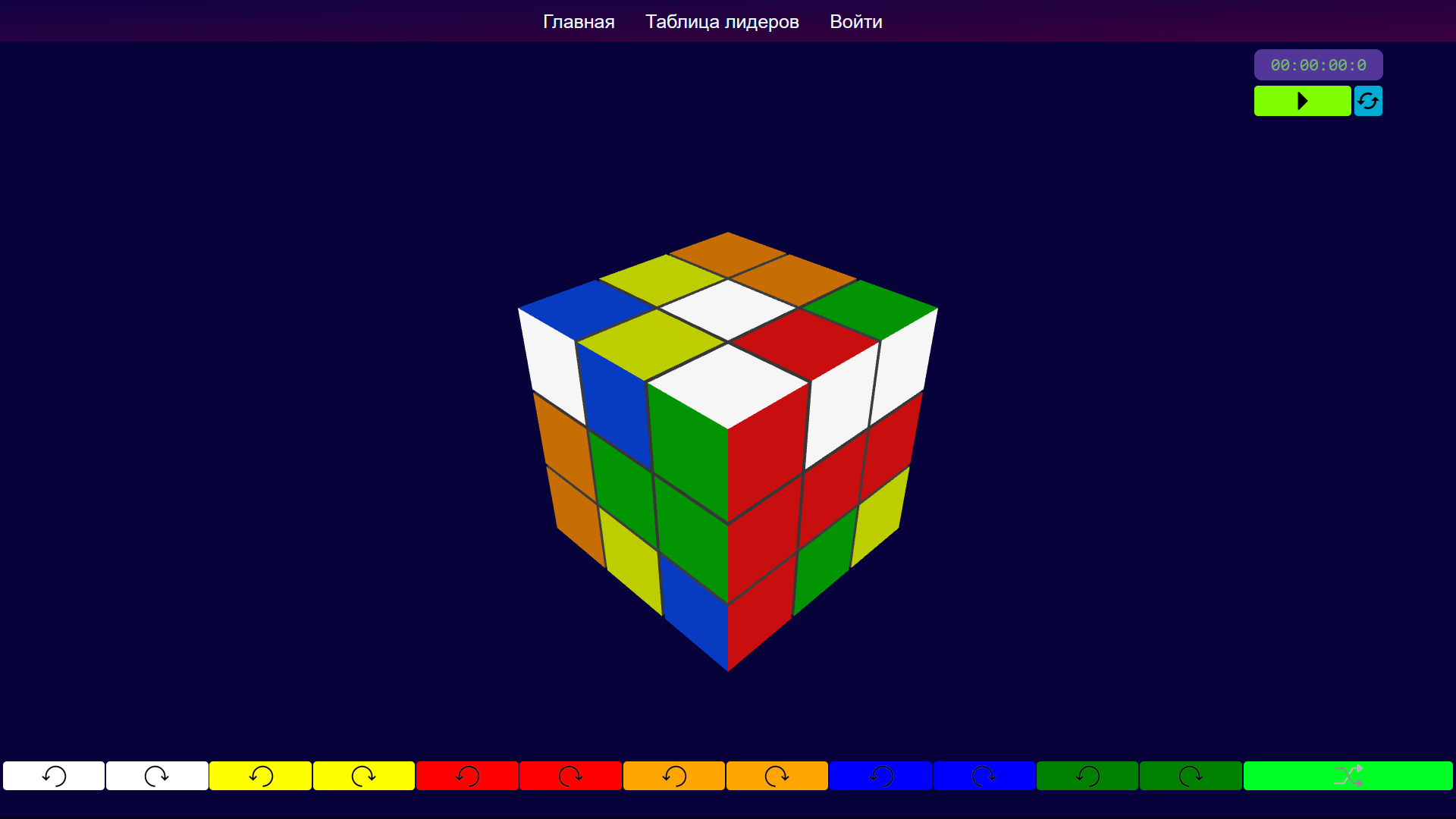


Рис. 2. Перемешанный кубик

При нажатии кнопок управления таймером, можно увидеть старт таймера

(рис. 3).

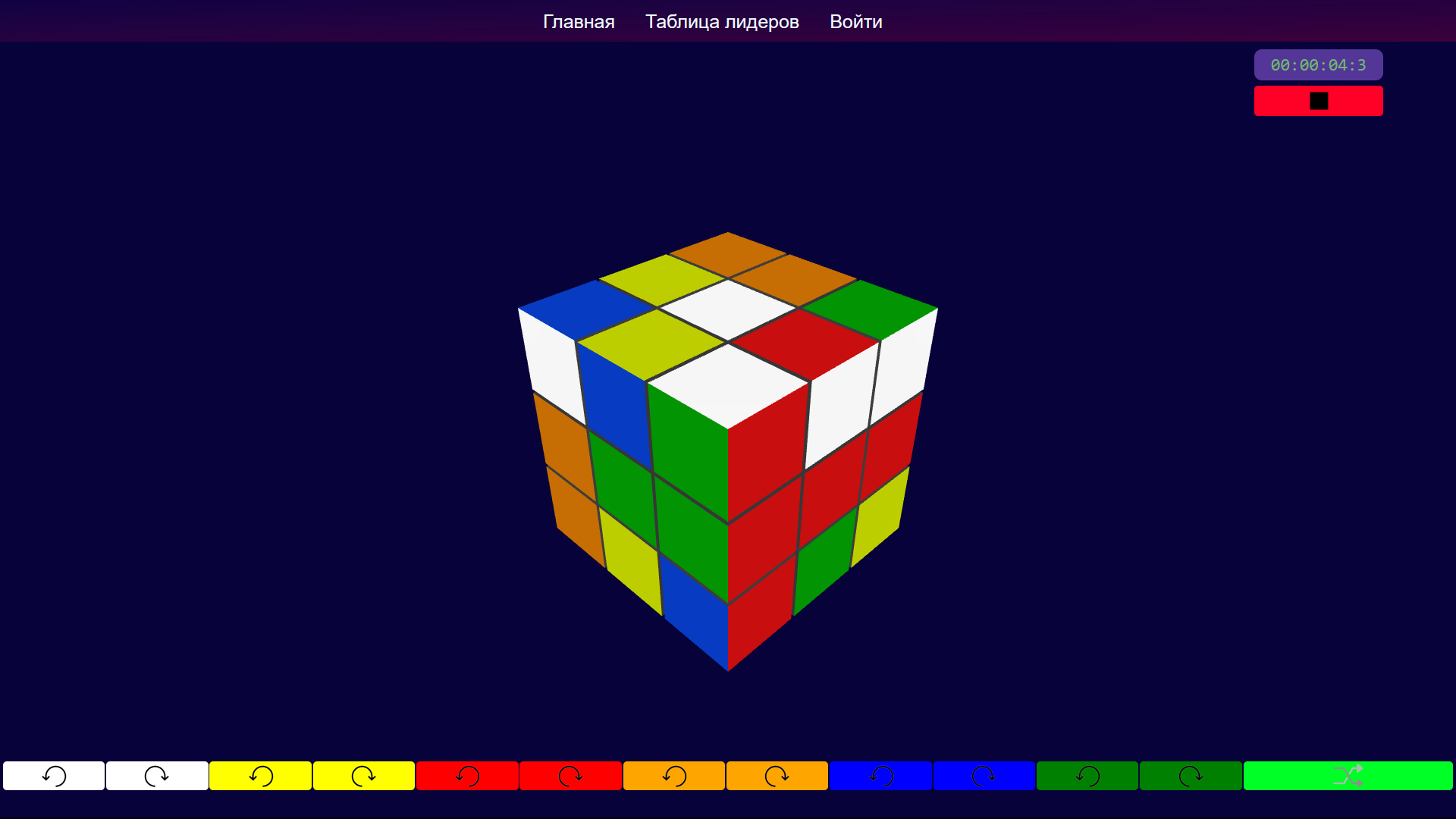


Рис. 3. Отсчет таймера

При нажатии кнопки «Войти» открывается страница входа (рис. 4).

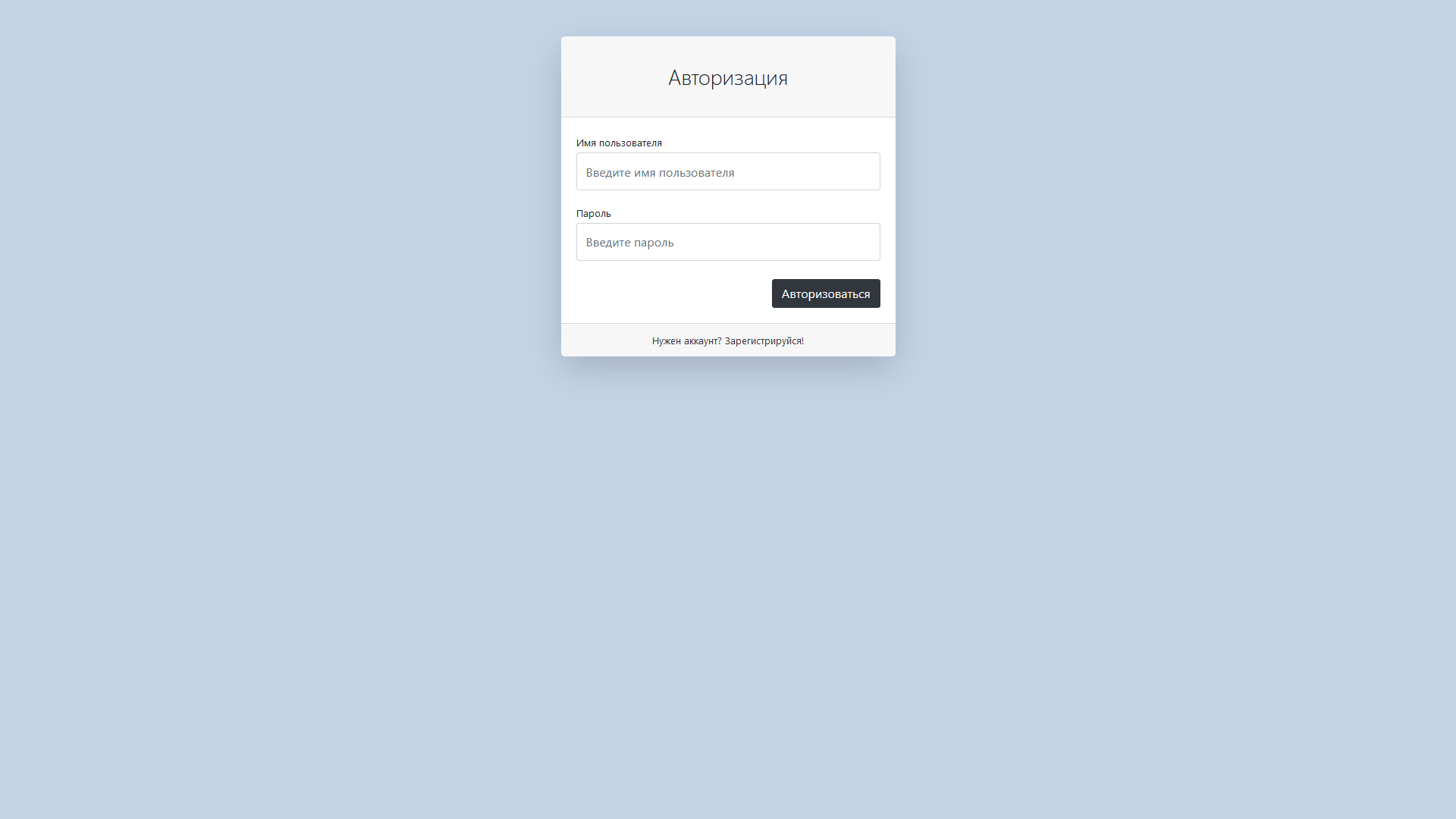


Рис. 4. Страница входа

В случае успешного входа, кнопка «Войти» изменится на кнопку «Выйти», а также в левом верхнем углу появится лучшее время сборки.

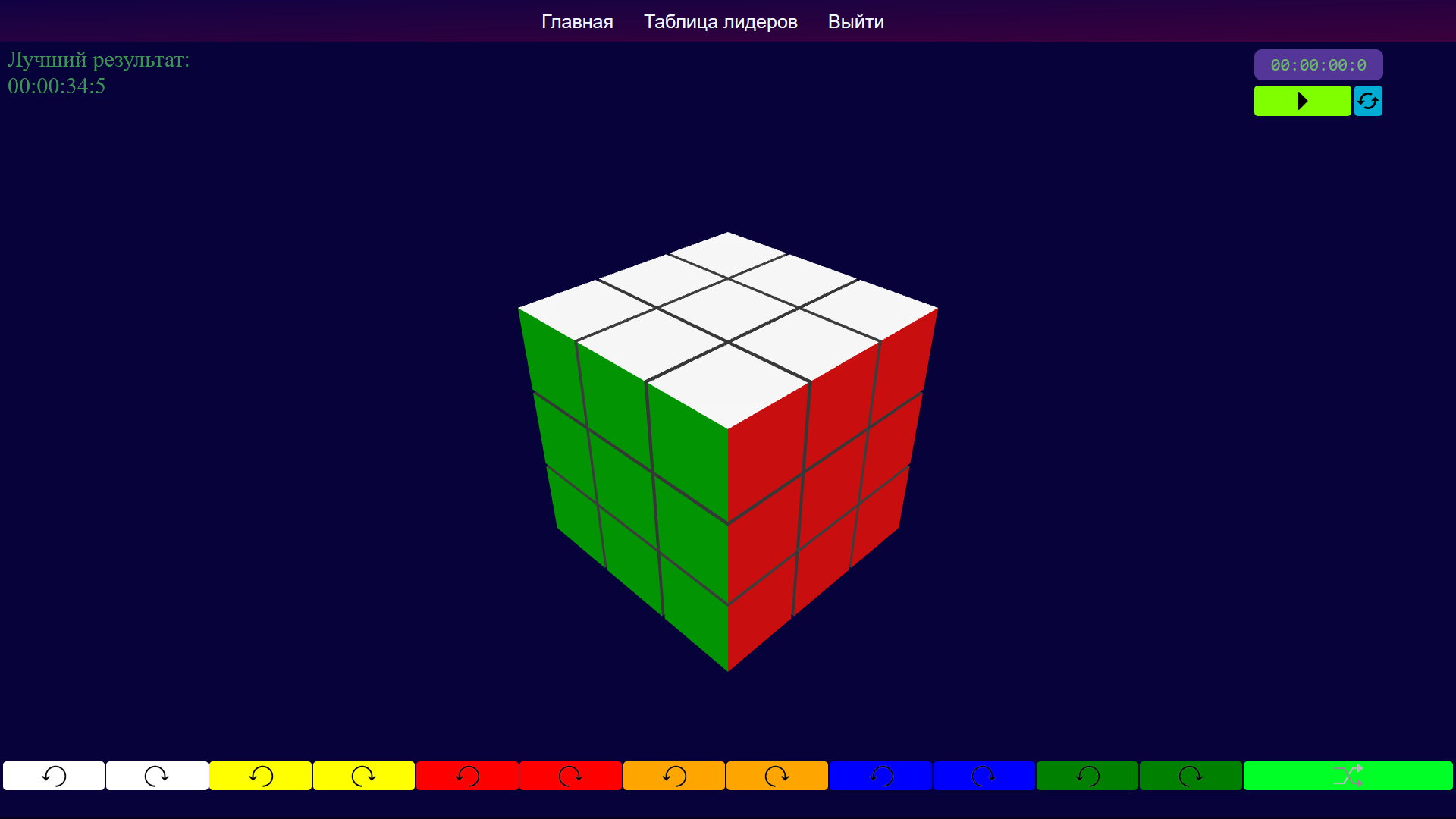


Рис. 5. Главная страница после авторизации

При нажатии кнопки «Таблица лидеров», открывается страница со списком (рис. 6).

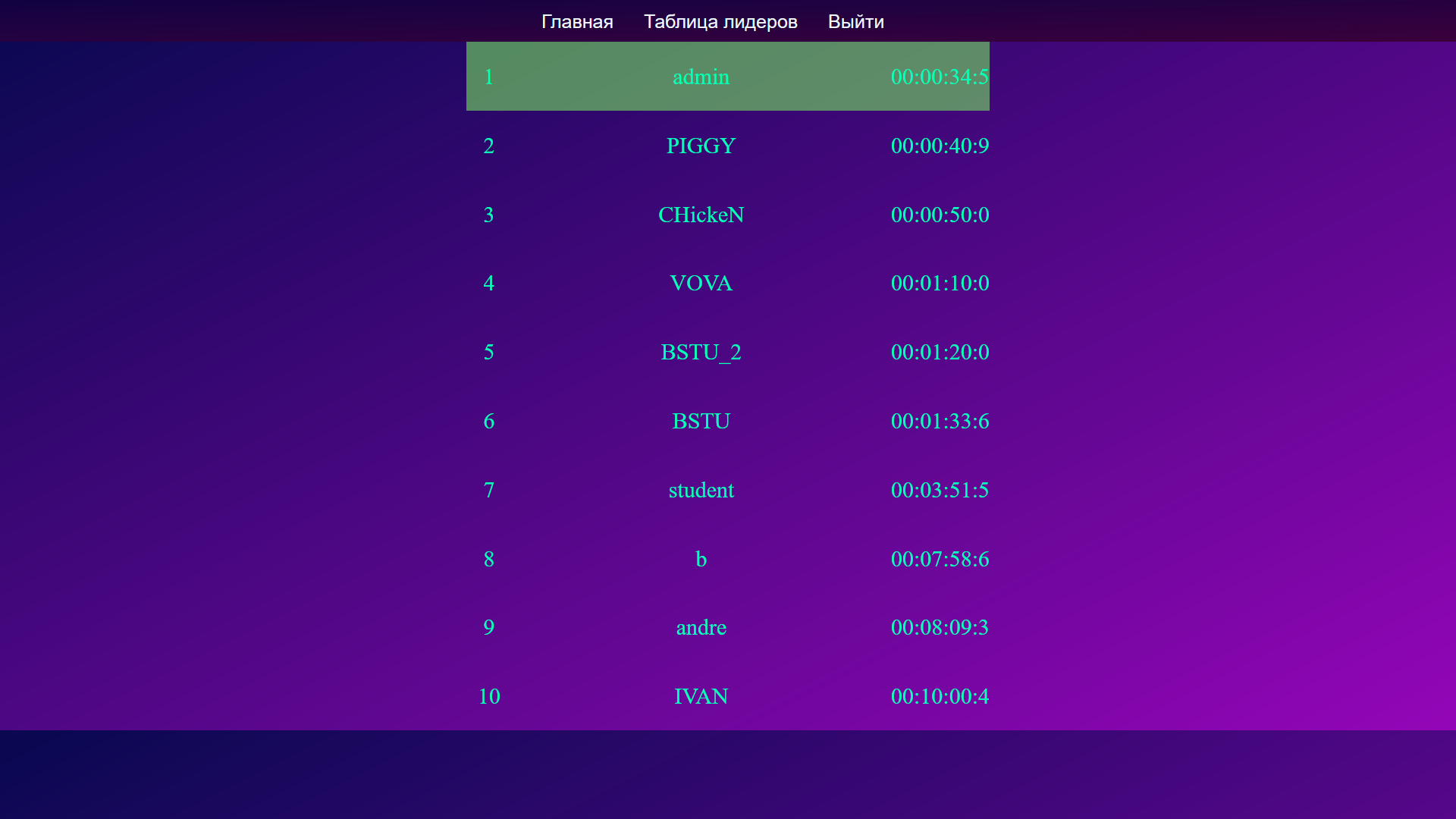


Рис. 6. Страница со списком лидеров

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной курсовой работы было разработано веб-приложение, предназначенное для сборки и решения головоломки - кубика Рубика. Целью работы было создание интуитивно понятного и функционального приложения, способного предоставить пользователям возможность виртуально собирать кубик Рубика и получать подсказки в процессе. В процессе разработки были использованы современные инструменты, такие как Python, HTML, CSS и JavaScript. Было проведено проектирование архитектуры и интерфейса приложения, учитывая удобство использования и эстетические аспекты. Основной функционал веб-приложения включает возможность вращения граней кубика Рубика, отслеживание изменения положения элементов. В результате работы было достигнуто поставленное целью создание полнофункционального веб-приложения по сборке кубика Рубика. Приложение успешно прошло тестирование и демонстрирует стабильную работу в различных веб-браузерах. Оно предоставляет пользователям интерактивный и увлекательный опыт сборки кубика Рубика в виртуальной среде.

Однако, несмотря на достигнутый результат, есть возможности для дальнейшего улучшения приложения. Например, можно реализовать дополнительные функции, такие как сохранение и загрузка состояния кубика, соревнования между пользователями и интеграцию с социальными сетями для обмена результатами и опытом.

В заключение, разработка веб-приложения по сборке кубика Рубика была интересной и продуктивной задачей. Приложение предоставляет пользователям возможность насладиться процессом сборки кубика Рубика, развивая логическое мышление и улучшая навыки решения головоломок.

Веб-приложение по сборке кубика Рубика имеет потенциал для применения в различных сферах. Например, оно может использоваться в образовательных целях, помогая студентам и учащимся развивать логическое мышление и пространственную ориентацию. Также оно может быть интересным развлечением для любителей головоломок, которые могут соревноваться с друзьями в сборке кубика Рубика через приложение.

Дальнейшее развитие приложения может включать добавление новых функций и улучшение существующих. Например, можно добавить режимы игры, решения кубика в ограниченное количество ходов или создания случайных комбинаций для большего разнообразия.

В целом, разработка веб-приложения по сборке кубика Рубика представляет собой интересную и актуальную задачу, объединяющую программирование, дизайн и логическое мышление. Результаты работы в виде полнофункционального приложения позволяют пользователям насладиться процессом сборки кубика Рубика в виртуальной среде, исключая необходимость в физическом кубике.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Введение в 3D: основы Three.js. [Электронный ресурс] — Режим доступа: hhttps://habr.com/ru/articles/494810/ — Дата доступа: 21.04.2023.
2. Веб-приложения: особенности, преимущества и отличия. [Электронный ресурс] — https://in-news.ru/news/veb-prilozeniia-osobennosti-preimushhestva-i-otliciia. — Дата доступа: 21.04.2023.
3. Громадская, О. А. Кубик Рубика — не просто развлечение! / О. А. Громадская, О. В. Мельничук. —// Юный ученый. — 2019. — № 7.2 (27.2). — [Электронный ресурс] — Режим доступа: URL: https://moluch.ru/young/archive/27/1624/ — Дата доступа:21.05.2023.
4. Елкина В. В мире продано 400 млн кубиков Рубика. [Электронный ресурс] — Режим доступа: https://rb.ru/story/rubiks-cube/ — Дата доступа: 21.04.2023.
5. Как оживить свой сайт: 5 инструментов для анимации в современном WEB. [Электронный ресурс] — Режим доступа: https://tproger.ru/digest/5-animation-tools-for-web/ — Дата доступа: 21.04.2023.
6. Lodash для начинающих. [Электронный ресурс] — Режим доступа: https://monsterlessons.com/project/series/lodash-dlya-nachinayushih — Дата доступа: 21.05.2023.
7. Математика кубика Рубика. [Электронный ресурс] — https://ru.wikipedia.org/wiki/Математика\_кубика\_Рубика — Дата доступа: 21.05.2023.
8. Почему Django — лучший фреймворк для разработки сайтов. [Электронный ресурс] — Режим доступа: https://ru.hexlet.io/blog/posts/pochemu-django-luchshiy-freymvork-dlya-razrabotki-saytov — Дата доступа: 21.04.2023.
9. Расширение модели пользователя в Django: сравнение нескольких стратегий с примерами кода. [Электронный ресурс] — Режим доступа: https://tproger.ru/translations/extending-django-user-model/ — Дата доступа: 21.05.2023.
10. Рейтинг языков программирования 2023. JavaScript/TypeScript завоевывают мир, Python вошел в топ-3. [Электронный ресурс] — Режим доступа: https://habr.com/ru/articles/730954/ — Дата доступа: 21.04.2023.
11. Создание первого приложения. [Электронный ресурс] — Режим доступа: https://metanit.com/python/django/1.4.php — Дата доступа: 21.04.2023.
12. Теорема Кэли. [Электронный ресурс] — Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=Теорема\_Кэли — Дата доступа: 21.04.2023.
13. Теорема Лагранжа (теория групп). [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Теорема\_Лагранжа\_(теория\_групп) — Дата доступа: 21.04.2023.
14. Теория групп. [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Теория\_групп — Дата доступа: 21.04.2023.
15. Удивительная математика внутри кубика Рубика. [Электронный ресурс] — Режим доступа: https://habr.com/ru/companies/skillfactory/articles/546094/ — Дата доступа: 21.04.2023.
16. Что на самом деле происходит, когда пользователь вбивает в браузер адрес google.com. [Электронный ресурс] — Режим доступа: https://habr.com/ru/companies/htmlacademy/articles/254825/ — Дата доступа: 21.04.2023.
17. Что такое JavaScript? [Электронный ресурс] — Режим доступа: https://aws.amazon.com/ru/what-is/javascript/ — Дата доступа: 21.04.2023.

**ПРИЛОЖЕНИЕ А**

**Инструкция по установке и запуске сервера**

Для запуска приложения необходимо иметь следующие программы/пакеты:

* Python (версии 3.6 или выше)
* Git
* Веб-браузер

Инструкция по запуску сервера:

1. Склонировать репозиторий проекта с GitHub:

git clone https://github.com/4eburaWka/Rubicks\_cube.git

1. Перейти в скачанную папку проекта:

cd Rubicks-cube

1. Установить все необходимые библиотеки:

pip install -r requirements.txt

1. Создать файл миграции:

python manage.py makemigrations

1. Применить миграции к базе данных:

python manage.py migrate

1. Произвести запуск приложения командой:

python manage.py runserver

**ПРИЛОЖЕНИЕ Б**

**Код проекта**

Код всего проекта размещен в репозитории GitHub. Его можно найти по ссылке https://github.com/4eburaWka/Rubicks\_cube.git