--------------------------------------------

## Общ преглед на Приложението

\*\*Доброволци БГ\*\* е уеб приложение, което свързва доброволци с организатори на инициативи. Идеята е потребителите-доброволци да могат да разглеждат различни инициативи, да кандидатстват за тях и да участват. Организаторите могат да публикуват нови инициативи, да одобряват кандидатури на доброволци и да управляват цялостния процес. Администраторът (admin) има най-широки права – може да одобрява инициативи, да управлява потребители, да вижда метрики и т.н.

Приложението се състои от два основни компонента:

1. \*\*Frontend\*\* – Създаден с React, React Router, Bootstrap, Axios и др. Отговаря за визуалната част и взаимодействието с потребителя.

2. \*\*Backend\*\* – Изграден с Node.js, Express.js, Sequelize и PostgreSQL. Осигурява REST API за данните, потребителските операции, инициативи, кандидатури и т.н. Съдържа моделите, маршрути, контролери, middleware и др.

--------------------------------------------

## Използвани Технологии

### Фронтенд:

- \*\*React\*\*: Използва се за изграждане на потребителския интерфейс.

- \*\*React Router\*\*: За маршрутизиране и навигация между страниците.

- \*\*Axios\*\*: За HTTP заявки към сървъра.

- \*\*Bootstrap и React-Bootstrap\*\*: За стилове и компоненти на интерфейса.

- \*\*Formik и Yup\*\*: За валидация на форми.

- \*\*Google Maps API\*\*: За визуализиране и избор на локации на карта.

- \*\*JWT (JSON Web Tokens)\*\*: Във фронтенда се съхраняват Access Token и Refresh Token за сигурна комуникация с бекенда.

### Бекенд:

- \*\*Node.js и Express.js\*\*: Сървърна рамка.

- \*\*Sequelize\*\*: ORM за работа с PostgreSQL база данни.

- \*\*JWT\*\*: За удостоверяване и авторизация на потребителите.

- \*\*Multer\*\*: За качване на файлове (изображения).

- \*\*bcryptjs\*\*: За хеширане на пароли.

- \*\*dotenv\*\*: За управление на environment-променливи.

- \*\*CORS\*\*, \*\*morgan\*\*: За логиране на заявки и управление на Cross-Origin достъп.

### База Данни:

- \*\*PostgreSQL\*\*: Релационна база данни.

- \*\*Sequelize Migrations\*\*: За управление на схемата на базата данни.

--------------------------------------------

## Структура на Проекта

### Фронтенд (папка `frontend`):

- `src/` – основната папка с React кода.

- `components/` – компонентите (Navbar, Footer, InitiativeCard, ApplicationsList и др.)

- `pages/` – страници (HomePage, LoginPage, RegisterPage, Dashboard, Profile, InitiativeDetail и т.н.)

- `contexts/` – React Context за управление на аутентикацията и зареждането (AuthContext, LoadingContext).

- `utils/` – помощни файлове (например `tokenUtils.js` за проверка на токени).

- `App.js` – главен компонент, който дефинира маршрути на фронтенда.

- `index.js` – входна точка на фронтенда.

### Бекенд (папка `backend`):

- `server.js` – входна точка на сървъра, конфигурация на Express, CORS, маршрути.

- `config/` – конфигурационни файлове и настройки за базата данни.

- `controllers/` – контролери, в които се реализира бизнес логиката при обработка на заявки:

- `authController.js`, `userController.js`, `initiativeController.js`, `applicationController.js`, `commentController.js`, `adminController.js`

- `models/` – Sequelize модели, описващи таблиците в базата данни:

- `User.js`, `Initiative.js`, `Application.js`, `Comment.js`, `Token.js`

- `routes/` – Express маршрути:

- `authRoutes.js`, `userRoutes.js`, `initiativeRoutes.js`, `applicationRoutes.js`, `adminRoutes.js`, `commentRoutes.js`

- `middleware/` – middleware функции:

- `auth.js` (защитен достъп), `authorize.js` (ролеви базирана авторизация), `upload.js`, `optionalAuth.js`

- `migrations/` – миграции за базата.

- `utils/` – спомагателни утилити (например `errorHandler.js` за глобална обработка на грешки).

--------------------------------------------

## Функционалности

### Потребители и Роли

- \*\*Роли\*\*: `volunteer` (доброволец), `organizer` (организатор), `admin` (администратор).

- \*\*Регистрация\*\*: Всеки може да се регистрира като доброволец или организатор.

- \*\*Вход\*\*: Генерира се `accessToken` (валиден 1 час) и `refreshToken` (валиден 7 дни).

- \*\*Профил\*\*: Потребителят може да обнови името, имейла и профилната си снимка, както и да смени паролата.

- \*\*Изход\*\*: Инвалидиране на `refreshToken`.

### Инициативи

- \*\*Създаване на инициатива\*\* (само от организатор или админ):

- Попълване на заглавие, описание, дата, категория, адрес, координати (чрез карта) и опционално изображение.

- \*\*Преглед на инициативи\*\*:

- Неаутентифицирани и доброволци виждат само одобрени инициативи.

- Организатори виждат своите (включително неодобрени), както и одобрените на други.

- Админът вижда всички.

- \*\*Одобряване на инициатива\*\* (само админ): Маркиране на инициативата като одобрена.

- \*\*Редактиране / Изтриване на инициатива\*\*: От организатора, който я е създал, или от админ.

### Кандидатури (Applications)

- \*\*Доброволци кандидатстват\*\* за инициативи: Създава се запис в таблицата `Application` със статус "Pending".

- \*\*Организатори/Админ\*\* виждат кандидатурите за техните инициативи и могат да ги одобряват или отхвърлят.

- Доброволец може да изтрие своя кандидатура, ако още не е одобрена или отхвърлена.

- Админ може да изтрие кандидатура директно.

### Коментари

- \*\*Добавяне на коментар\*\*: Доброволец, организатор или админ могат да добавят коментар към инициатива.

- \*\*Преглед на коментари\*\*: Всички (включително неаутентифицирани) могат да прочетат коментарите.

- \*\*Изтриване на коментар\*\*: Само авторът на коментара или админът могат да го изтрият.

### Администратор (Admin)

- Управление на потребители: Виждане на списък с всички потребители, изтриване на потребители.

- Вижда и одобрява / отхвърля инициативи.

- Има достъп до метрики: общ брой потребители, инициативи, кандидатури, разбивка по статуси и роли.

### Карта (Google Maps API)

- При добавяне/редактиране на инициатива организаторът избира локация от карта.

- На началната страница се визуализират всички одобрени инициативи на карта.

- На страницата на конкретна инициатива има карта с локация на събитието.

--------------------------------------------

## Подготовка и Стартиране на Приложението

### Предварителни Изисквания

1. \*\*Node.js и npm\*\* трябва да са инсталирани.

2. \*\*PostgreSQL\*\* база данни трябва да е достъпна.

3. В `.env` файл трябва да се настроят следните променливи:

- `DB\_HOST`, `DB\_NAME`, `DB\_USERNAME`, `DB\_PASSWORD` – данни за връзка с базата данни.

- `JWT\_SECRET`, `JWT\_REFRESH\_SECRET` – секрети за подписване на JWT токени.

- `REACT\_APP\_GOOGLE\_MAPS\_API\_KEY` – API ключ за Google Maps.

Примерно съдържание на `.env`:

```

DB\_HOST=localhost

DB\_NAME=dobrovoltsi\_db

DB\_USERNAME=postgres

DB\_PASSWORD=secret

JWT\_SECRET=supersecretkey

JWT\_REFRESH\_SECRET=refreshsupersecretkey

REACT\_APP\_GOOGLE\_MAPS\_API\_KEY=YOUR\_GOOGLE\_MAPS\_API\_KEY

```

### Инсталиране на зависимости

- Фронтенд:

```

cd frontend

npm install

```

- Бекенд:

```

cd backend

npm install

```

### Стартиране на Базата Данни

Уверете се, че PostgreSQL е стартиран и имате създадена база данни с име, съответстващо на `DB\_NAME` от `.env`.

### Стартиране на Бекенда

```

cd backend

node server.js

```

Сървърът по подразбиране слуша на порт 5000. Ще бъде създаден администраторски потребител (ако не съществува) с:

- email: administrator@dobrovoltsibg.com

- password: dobrovoltsibg

### Стартиране на Фронтенда

```

cd frontend

npm start

```

Фронтендът по подразбиране слуша на порт 3000, ще се отвори на `http://localhost:3000`.

--------------------------------------------

## Архитектура на Бекенда

### Модели

- \*\*User\*\*: Потребител с полета `name`, `email`, `password`, `role`, `profileImage`.

- \*\*Initiative\*\*: Инициатива с полета `title`, `description`, `date`, `category`, `address`, `latitude`, `longitude`, `imageUrl`, `organizerId`, `approved`.

- \*\*Application\*\*: Кандидатура, свързана с `initiativeId` и `volunteerId`, полето `status`.

- \*\*Comment\*\*: Коментар, свързан с `initiativeId` и `userId`.

- \*\*Token\*\*: Refresh токени за потребители.

### Контролери

- \*\*authController\*\*: Регистрация, вход, refresh token, logout, getProfile.

- \*\*userController\*\*: Преглед и обновяване на профил, смяна на парола, изтриване на акаунт, админ функционалности за потребители.

- \*\*initiativeController\*\*: CRUD операции по инициативи, одобряване на инициативи.

- \*\*applicationController\*\*: Кандидатстване, преглед и одобряване/отхвърляне на кандидатури.

- \*\*commentController\*\*: Добавяне, преглед и изтриване на коментари.

- \*\*adminController\*\*: Метрики и статистики за админ.

### Маршрути

- `/api/auth` – регистрация, вход, refresh, logout, профил.

- `/api/users` – управление на личен профил, смяна на парола, триене на акаунт.

- `/api/initiatives` – CRUD за инициативи, коментари към тях.

- `/api/applications` – кандидатстване и управление на кандидатури.

- `/api/admin` – администраторски маршрути (потребители, статистики, неодобрени инициативи).

- `/api/comments` – изтриване на коментари.

### Middleware

- \*\*protect\*\*: Изисква валиден JWT Access Token.

- \*\*authorize\*\*: Проверява роля на потребителя.

- \*\*optionalAuth\*\*: Позволява заявка без токен (ако има токен, разпознава потребителя).

- \*\*upload\*\*: Обработка на качване на файлове с Multer.

- \*\*errorHandler\*\*: Глобален обработчик на грешки.

--------------------------------------------

## Архитектура на Фронтенда

### Основни Страници

- \*\*HomePage\*\*: Начална страница, кратка информация, бутони за вход/регистрация или преглед на инициатви, карта с всички инициативи.

- \*\*LoginPage / RegisterPage\*\*: Форми за вход и регистрация.

- \*\*Dashboard\*\*:

- За доброволец: Показва кандидатурите, за които е кандидатствал.

- За организатор: Показва създадените инициативи и кандидатурите към тях.

- \*\*AddInitiative / EditInitiative\*\*: Добавяне и редактиране на инициатива (само от организатор/админ).

- \*\*InitiativeList\*\*: Списък с всички одобрени инициативи (или достъпни според ролята).

- \*\*InitiativeDetail\*\*: Детайли за избрана инициатива, опция за кандидатстване (за доброволец), коментари, карта с локация.

- \*\*Profile / ProfileEdit / ChangePassword\*\*: Преглед на профил, редактиране на профил, смяна на парола.

- \*\*AdminDashboard\*\*: (За админ) Показва метрики, управление на потребители и инициативи.

- \*\*AdminUsers\*\*: Списък с потребители, изтриване на потребител.

- \*\*NotFound\*\*: За невалидни маршрути.

### Компоненти

- \*\*Navbar\*\*, \*\*Footer\*\*: Универсални части на интерфейса.

- \*\*InitiativeCard\*\*, \*\*ApplicationsList\*\*, \*\*InitiativesMap\*\*: Визуални компоненти за конкретни части от функционалността.

- \*\*LoadingOverlay\*\*: Показва spinner при зареждане.

- \*\*LocationPicker\*\*: Компонент с Google Maps за избор на координати.

- \*\*PrivateRoute\*\*: Ограничен достъп до страници според потребителската роля и автентикация.

### Context

- \*\*AuthContext\*\*: Управлява състоянието на аутентикация, токени, refresh логика.

- \*\*LoadingContext\*\*: Управлява глобалния loading overlay при заявки.

--------------------------------------------

## Поток на Работа на Приложението

1. \*\*Потребител (неаутентифициран)\*\* влиза в `HomePage`, може да се регистрира или логне.

2. При \*\*регистрация\*\* или \*\*вход\*\* се получава `accessToken` и `refreshToken`, които се съхраняват в `localStorage` на браузъра. След това Axios изпраща токена в `Authorization` header за всяка защитена заявка.

3. \*\*Доброволец\*\*:

- Може да разглежда одобрени инициативи, да кандидатства за тях, да изтрива свои кандидатури, да добавя коментари.

- Има достъп до своя `Dashboard`, където вижда подадените кандидатури.

4. \*\*Организатор\*\*:

- Може да добавя инициативи, които първо са неодобрени.

- В своя `Dashboard` вижда своите инициативи (одобрени или не) и кандидатурите към тях.

- Може да преглежда и одобрява/отхвърля кандидатурите.

5. \*\*Администратор\*\*:

- Има `AdminDashboard`, където вижда статистики и метрики.

- Може да одобрява инициативи, да изтрива потребители, да разглежда всички инициативи и потребители.

- Има пълен контрол върху съдържанието.

--------------------------------------------

## Допълнителни Бележки

- \*\*Сигурност на Токени\*\*: Access Token изтича след 1 час, Refresh Token след 7 дни. При изтичане на Access Token, фронтендът се опитва да го обнови чрез Refresh Token. Ако Refresh Token също е невалиден или изтекъл, потребителят се разлогва.

- \*\*Качване на Изображения\*\*: Използва се Multer. Изображенията се съхраняват в папка `uploads/`. При рендиране на данни се връща пълен URL до изображението.

- \*\*Грешки\*\*: Има глобален error handler на бекенда. При 401 или 403 отговори, фронтендът пренасочва потребителя или показва съответно съобщение.

--------------------------------------------

## Заключение

Това приложение е комплексна платформа за взаимодействие между доброволци, организатори и администратори. Чрез ясни модели, структуриран бекенд и добре организиран фронтенд се предлага сигурен, модулен и лесно разширяем софтуер. Документацията по-горе има за цел да предостави цялостна картина на кода, функционалностите и начина на работа, като улесни бъдещото развитие, поддръжка и интеграция на нови функционалности.

--------------------------------------------

## Сценарий за Неаутентифициран Потребител (Гост)

\*\*Цел\*\*: Гостът разглежда публичната информация на платформата без да е вписан.

1. \*\*Отваря началната страница (`HomePage`)\*\*:

- Вижда обща информация, бутони за вход/регистрация, и карта с инициативи (само одобрени).

- Може да види списък с инициативи (`InitiativeList`), но без детайли за кандидатстване.

2. \*\*Преглежда Инициативи\*\*:

- Навигира до `/initiatives`.

- Вижда списък само от одобрени инициативи.

- Може да кликне на някоя инициатива и да види детайлите (`InitiativeDetail`) – заглавие, описание, дата, карта с локация, коментари.

- Вижда и коментарите, но не може да коментира без да се логне.

3. \*\*Решава да се регистрира или впише\*\*:

- Отива на `/login` или `/register`.

- Ако се регистрира като доброволец или организатор, след успешна регистрация се логва автоматично.

- Ако се логне успешно, става автентифициран и вече има достъп до допълнителни функционалности.

--------------------------------------------

## Сценарий за Доброволец (Volunteеr)

\*\*Цел\*\*: Доброволецът разглежда инициативи и кандидатства за участие.

1. \*\*Вход в системата\*\*:

- Навигира до `/login`, въвежда email и парола.

- След успешен вход бива пренасочен към `/` или `/dashboard`.

2. \*\*Преглед на Инициативи като Доброволец\*\*:

- От менюто клика върху "Инициативи" (`/initiatives`).

- Вижда списък с одобрени инициативи.

- Клика на конкретна инициатива (`/initiatives/:id`) за повече детайли.

3. \*\*Кандидатстване за Инициатива\*\*:

- На страницата с детайлите за инициативата вижда бутон "Кандидатстване" (ако не е кандидатствал досега).

- Натиска бутона и се подава кандидатура. Статусът е `Pending`.

4. \*\*Преглед на Кандидатурите си\*\*:

- Отива в своя `Dashboard` (`/dashboard`).

- Вижда списък с кандидатурите си, техния статус (Pending/Approved/Denied).

- Ако кандидатурата е все още в статус Pending, може да я изтрие.

5. \*\*Коментари\*\*:

- На страницата на конкретна инициатива може да добави коментар, защото е логнат.

- Може да изтрие свой собствен коментар по-късно.

6. \*\*Редакция на Профила\*\*:

- Отива в `/profile`, вижда своя профил.

- Може да редактира профила (`/profile/edit`) – име, имейл, снимка.

- Може да смени паролата (`/profile/change-password`).

--------------------------------------------

## Сценарий за Организатор (Organizer)

\*\*Цел\*\*: Организаторът добавя и управлява инициативи, преглежда кандидатури от доброволци.

1. \*\*Вход като Организатор\*\*:

- Навигира до `/login`, въвежда email и парола.

- След вход, ако е организатор, може да достъпи `/dashboard`.

2. \*\*Добавяне на Нова Инициатива\*\*:

- Отива в `/initiatives/add`.

- Попълва заглавие, описание, дата, категория.

- Избира локация от картата.

- Качва снимка (опционално).

- Запазва инициативата.

- Инициативата се създава с `approved = false` и трябва да бъде одобрена от админ.

3. \*\*Преглед на Собствените Инициативи\*\*:

- В `Dashboard` вижда списъка със своите инициативи.

- Може да редактира или изтрие своя инициатива, докато е неодобрена или одобрена.

- Одобрени инициативи се виждат от доброволци, неодобрените – не, но организаторът ги вижда.

4. \*\*Преглед на Кандидатури към Неговите Инициативи\*\*:

- В `Dashboard` вижда таблица с кандидатури за инициативите, които е създал.

- Може да одобрява (`Approved`) или отхвърля (`Denied`) кандидатури.

- Вижда името, имейла на доброволеца, и заглавието на инициативата, за която е кандидатствал.

5. \*\*Коментари\*\*:

- Може да оставя коментари към своите или други одобрени инициативи.

- Може да изтрива собствените си коментари.

6. \*\*Профилни Настройки\*\*:

- Подобно на доброволеца, може да редактира профил и да смени парола.

--------------------------------------------

## Сценарий за Администратор (Admin)

\*\*Цел\*\*: Администраторът одобрява инициативи, управлява потребители, вижда метрики.

1. \*\*Вход като Администратор\*\*:

- Логва се с админските креденциали.

- След успех, има достъп до `/admin` (Admin Dashboard).

2. \*\*Преглед и Одобрение на Инициативи\*\*:

- В `/admin` вижда списък с всички инициативи, включително неодобрени.

- За неодобрени инициативи има бутон "Одобряване".

- След като одобри, инициативата вече се вижда от доброволци и неаутентифицирани потребители.

3. \*\*Управление на Потребители\*\*:

- От `/admin/users` вижда списък с всички потребители.

- Може да изтрие потребител (например ако е фалшив или нарушител).

4. \*\*Метрики\*\*:

- В `Admin Dashboard` вижда метрики: брой потребители, инициативи, кандидатури, разбивки по статус.

- Например, графики с PieChart: потребители по роли, инициативи одобрени/неодобрени, кандидатури по статус.

5. \*\*Коментари\*\*:

- Като админ може да изтрива коментари на други потребители, ако са неподходящи.

6. \*\*Организаторски Функционалности\*\*:

- Админът има всички права на организатор и доброволец.

- Може да създава, редактира, триe инициативи, както и да кандидатства (макар това да не е типично за админ).

--------------------------------------------

## Обобщение на примерните схеми

- \*\*Гост\*\*: Преглежда публична информация, вижда само одобрени инициативи, не може да кандидатства или коментира, докато не се логне.

- \*\*Доброволец\*\*: Регистрира се, логва се, разглежда одобрени инициативи, кандидатства за тях, пише коментари, следи статусите си през Dashboard.

- \*\*Организатор\*\*: Създава инициативи, очаква одобрение от админ, след одобрение има доброволци, които кандидатстват, организаторът решава кои да одобри. Редактира и трие свои инициативи, контролира коментари (само своите).

- \*\*Администратор\*\*: Най-високо ниво на достъп. Управлява всички потребители и инициативи. Вижда метрики, одобрява инициативи, изтрива потребители, контролира съдържанието. Има достъп и до всички останали действия поради по-широките си права.

Тези сценарии представят типичния работен поток и дават ясна представа за възможностите, които всяка роля има след като влезе в системата.