

# Backend Challenge Spring 2024

## Blueprint Spring 2024 Backend Challenge

This challenge is an opportunity for you to showcase your skills, creativity, and understanding of backend technologies. This challenge was designed to simulate real-world problems that we had to solve in our previous NPO projects.

### Context and Challenge Overview

Backend systems are the backbone of any web application. As a backend developer, your task is to ensure that the server, application, and database communicate with each other safely and efficiently. The core of a successful backend is its ability to perform CRUD (Create, Read, Update, Delete) operations effectively, as these operations are fundamental to interacting with data stored in databases.

You will need *git* installed in order to work on this project. You can read this [Blueprint Developer Guide](#) for details on how to download and install git. You will also need to make an account on [GitHub](#) in order to access the code repository.

### Step 1: Fork the Repository

The first step is to fork the repository that contains the Frontend Challenge starter code. This creates a copy of the repository in your own GitHub account that you can work on without affecting the original.

To fork the repository:

1. Navigate to the repository on GitHub ([https://github.com/stevensblueprint/backend\\_challenge\\_spring\\_2024](https://github.com/stevensblueprint/backend_challenge_spring_2024))

2. Click the "Fork" button in the top right corner of the page
3. Select your GitHub account as the destination for the fork

## Step 2: Clone the Repository

Next, you need to clone the repository to your local machine. This creates a local copy of the repository that you can make changes to.

To clone the repository:

1. Open a terminal window
2. Navigate to the directory where you want to clone the repository
3. Run the following command: `git clone <repository-url>` Replace `<repository-url>` with the URL of your forked repository. The repository URL can be found by pressing the green 'Code' button and copying the HTTPS link. It will be a github link that includes your github username and forked repo name.

## Step 3: Create a New Branch

Before you make changes to the HTML file, you should create a new branch. This allows you to make changes without affecting the main branch of the repository. All development work for our teams exists in branches and are later merged into main with Pull Requests!

To create a new branch:

1. Navigate to the cloned repository directory in your terminal
2. Run the following command: `git checkout -b feature/yourName` Replace `yourName` with your first name. So your full branch name could be `feature/miguel`.

# Challenge Details

Your task is to develop a CRUD application that adheres to the following specifications and requirements:

## Application Overview

- **Purpose:** Create a backend service that Creates, Reads, Updates, and Deletes volunteers from a database.
- **Technology Stack:** You are free to choose the technology stack (programming languages, database systems, frameworks) that you are most comfortable with.

## CRUD Operations

- **Create:** Implement functionality to add new records to your database. For example, adding a new volunteer.

- **Read:** Develop methods to retrieve and display records from the database. This could include listing all volunteers or searching for a specific user.
- **Update:** Allow for modifications to existing records, such as updating a volunteer's contact information.
- **Delete:** Provide a way to remove records from the database safely.

# Volunteers Table

Field Name	Data Type	Description	Example
volunteer_id	UUID / INT (PK)	Unique identifier for each volunteer.	123e4567- e89b- 12d3- a456- 426614174000
first_name	VARCHAR	Volunteer's first name.	John
last_name	VARCHAR	Volunteer's last name.	Doe
email	VARCHAR	Volunteer's email address.	john.doe@example.com
phone_number	VARCHAR	Volunteer's phone number.	+1234567890
date_of_birth	DATE	Volunteer's date of birth.	1990- 01- 01
address	VARCHAR	Volunteer's address.	123 Main St, Anytown, AS
skills	VARCHAR / ARRAY	List of volunteer skills.	[ "First Aid", "Counseling" ]
availability	JSON / VARCHAR	Volunteer's availability (could be structured as JSON).	{ "Monday": [ "AM" ], "Saturday": [ "PM" ] }
date_joined	DATE	Date when the volunteer joined.	2024- 03- 21
background_check	BOOLEAN	Indicates if a background check has been completed.	true / false

# Volunteers

- **List Volunteers**
  - **GET** /api/volunteers
    - Retrieves a list of all volunteers, with options for pagination, filtering, and sorting.
- **Create a New Volunteer**
  - **POST** /api/volunteers
    - Creates a new volunteer record.
- **Get Volunteer Details**
  - **GET** /api/volunteers/{volunteerId}
    - Retrieves detailed information about a specific volunteer.
- **Update Volunteer Information**

- **PUT** `/api/volunteers/{volunteerId}`
  - Updates an existing volunteer's information.
- **Delete a Volunteer**
  - **DELETE** `/api/volunteers/{volunteerId}`
    - Deletes a specific volunteer's record.

## Volunteer Skills (Optional)

- **List Skills for a Volunteer**
  - **GET** `/api/volunteers/{volunteerId}/skills`
    - Lists the skills associated with a specific volunteer.
- **Add Skills to a Volunteer**
  - **POST** `/api/volunteers/{volunteerId}/skills`
    - Adds new skills to a volunteer.
- **Remove a Skill from a Volunteer**
  - **DELETE** `/api/volunteers/{volunteerId}/skills/{skillId}`
    - Removes a specific skill from a volunteer.

## Event Participation (Optional)

- **List Events a Volunteer Participated In**
  - **GET** `/api/volunteers/{volunteerId}/events`
    - Lists events that a specific volunteer has participated in.
- **Add Volunteer Participation to an Event**
  - **POST** `/api/events/{eventId}/volunteers/{volunteerId}`
    - Records a volunteer's participation in a specific event.
- **Remove Volunteer Participation from an Event**
  - **DELETE** `/api/events/{eventId}/volunteers/{volunteerId}`
    - Removes a record of a volunteer's participation in a specific event.

## Submission

1. Commit your code to your branch on your forked repository.
2. Create a Pull Request (PR).
3. Email a link to your Pull Request (PR) to [sit.blueprint@gmail.com](mailto:sit.blueprint@gmail.com). Please make the subject of your email "Backend Challenge: YOURNAME" i.e "Backend Challenge: Miguel Merlin".

This challenge is due on March 28 @ 11:59 P.M. Challenges will be discussed during the 3rd round of applications: Coffee Chats. Thank you for your time!

# Submission Checklist

1. Add a README.md where you tell us how to install dependencies and how to run your application
2. Create a pull request in your own forked repository
3. Submit the link to your pull pull-request

---

Revision #5

Created 29 February 2024 22:13:18 by Miguel Merlin

Updated 3 October 2024 20:21:46 by Miguel Merlin