

# Fullstack Developer Case

Showcase your understanding of building APIs and CRUD operations

## Introduction:

You have been assigned this case to demonstrate your proficiency in working with APIs and CRUD operations. Over the next 72 hours, you'll have the chance to exhibit these skills by choosing 1 of the 3 options presented below. All 3 options share the same task descriptions and evaluation criteria.

**NOTE:** This case is not intended to be completed in its entirety but rather to assess how you set up the project, prioritize tasks, demonstrate your thought process, and execution. You do not need to focus on design or spend too much time on it, as we are primarily looking for functionality and your implementation of the requirements described below. Good luck!

## Option 1: Recipe Sharing Platform API

Build an API for a recipe sharing platform that allows users to create, view, and search for recipes. Design the API routes for handling recipes and search functionalities. Implement CRUD operations for recipes, and enable users to search for recipes based on ingredients or categories. You can fetch data from your own database, or fetch your data from an existing database, such as [TheMealDB API](#)

## Option 2: Task Management API

Create an API for a task management application that allows users to organize and track their tasks. Design the API routes for handling tasks and task

categorization. Implement CRUD operations for tasks, and enable users to mark tasks as completed and categorize them for better organization.

### Option 3: Polling Application API

Develop an API for a polling application that allows users to create and participate in polls. Design the API routes for handling polls and voting. Implement CRUD operations for polls, and enable users to vote on polls and view poll results - in real-time (optional).

### Task Description:

#### 1. Build the API Endpoints

- Create your application using [Create T3 App](https://create.t3.gg) (<https://create.t3.gg>)
- Use the following settings after running the `npm create t3-app@latest` command:
  - What will your project be called?
    - Name your application
  - Will you be using TypeScript or JavaScript?
    - TypeScript (It will use it regardless of what you pick)
  - Will you be using Tailwind CSS for styling?
    - Yes
  - Would you like to use tRPC?
    - Yes
  - What authentication provider would you like to use?
    - NextAuth (recommended), but you can choose whatever auth library you'd like instead.
    - **NOTE!** If you pick NextAuth, remember to setup your Provider Credentials (Defaults to discord - [check link](#))
  - What database ORM would you like to use?
    - Prisma
  - Would you like to use Next.js App Router?
    - Yes

*The rest of the questions are up to you to decide.*

- Design endpoints following RESTful principles and ensure proper handling of HTTP methods (GET, POST, PUT, DELETE).
2. Implement CRUD operations
    - Implement Create, Read, Update, and Delete (CRUD) operations.
    - Utilize Prisma ORM to interact with the database and ensure data consistency and integrity.
  3. Display fetched data
    - Use tRPC to both fetch the data and to mutate the data.
      - Although tRPC is preferred, you can also use fetching libraries like Fetch API, Axios, React Query or similar to fetch the data in alongside or instead of tRPC.
    - Develop frontend components using React to display data from the API unto the client side.
    - Render the data on appropriate pages with appropriate styling.
  4. Deploy application
    - Deploy the application to a production environment like Vercel or Netlify for testing and demonstration purposes.
    - Ensure the application runs smoothly in a live environment and handle any deployment-related issues.

**Requirements:**

- Use Next.js and TypeScript for the project.
- Utilize standardized API rules and follow best practices for code quality and structure.
- Create a user-friendly and intuitive user experience throughout the coding process.
- And lastly, have fun! 🎉

**Bonus:**

Do you want to stand out? Add auth functionality to your application (as described in the task description), and only allow authorized users to perform CRUD Operations.

**Assessment Criteria:**

- **Functionality:** How well does the solution meet the specified requirements and functionalities?
- **Code Quality:** How clean, structured, and reusable is the code?
- **User Interface:** How well is the user interface designed and adapted to the user experience?
- **Bonus:** Are there any additional features or improvements you add to enhance the experience?

You are expected to complete this case within 72 hours and submit your written responses via email to [zakaria@event.no](mailto:zakaria@event.no) - If you have any questions regarding this case assignment, feel free to contact us via email, and we will assist you to the best of our abilities. Good luck!