



Politechnika
Wrocławska

Bazy Danych II

Cooking_Blog

Janiszewski Gracjan
Muszyński Bartłomiej
Nowacki Aleksander
Szymczuk Mikołaj
Troszczyński Adam



HR EXCELLENCE IN RESEARCH



Cooking Blog

🔗 Authors:

Name	Role
Mikołaj Szymczuk	Frontend , Backend Support , Database Support , Testing support
Adam Troszczyński	Frontend , Database Support , Testing support
Gracjan Janiszewski	Testing (Unit, E2E, Integration, Finding bugs and problems with app), Database Support
Aleksander Nowacki	Backend , Database
Bartłomiej Muszyński	Backend , Database

About project

Cooking Blog is culinary blog app where users can add, explore and share dish recipes.

Project setup

For developing app

Recommended IDE - Visual Studio Code

Recommended extensions (Copy id and paste in extension search bar in VS Code):

Extension Name	Extension ID
EditorConfig	<code>EditorConfig.EditorConfig</code>
Eslint	<code>dbaeumer.vscode-eslint</code>
GitLens	<code>eamodio.gitlens</code>
PathIntellisense	<code>christian-kohler.path-intellisense</code>
Prettier	<code>esbenp.prettier-vscode</code>
Px to Rem	<code>cipchk.cssrem</code>
TailwindCSS	<code>bradlc.vscode-tailwindcss</code>
Todo Highlight	<code>wayou.vscode-todo-highlight</code>
Vue Typescript Support	<code>Vue.vscode-typescript-vue-plugin</code>
Vue Language Support	<code>Vue.volar</code>
Vue Snippets	<code>sdras.vue-vscode-snippets</code>

[Client]

Requirements:

- Node version ≥ 16

Go to client directory and install all dependencies

```
cd client  
npm install
```



Next you can run project and develop app

```
npm run dev
```



To run unit tests

```
npm run test:unit
```

To run E2E (End-To-End) first you need to build client app

```
npm run build
```



and then you can run tests

and then you can run tests

```
npm run test:e2e
```



🔗 [Database]

Requirements:

- Postgresql server
- Some client for Postgresql ([DataGrip](#), [PgAdmin](#))

Install PostgreSQL server

Then create two databases `cookingBlog` and `cookingBlog_test`

For example on postgres account you can type

```
createdb cookingBlog  
createdb cookingBlog_test
```



Or you can create databases in PgAdmin

Next you need to create all tables structure, in your postgresql client or by using terminal run script

```
cookingBlogInit.sql
```

At the end you can run insert script `insertBasicData.sql` to insert some testing data to database

[Server]

Requirements:

- Node version ≥ 16

Go to server directory and install all dependencies

```
cd server  
npm install
```



Create `.env` file with all environment variables (port 8080 is important because client send requests to server with this port so try to not change it)

```
PORT=8080  
DB_USER=[user]  
DB_HOST=[host]  
DB_NAME=[database name]  
DB_PASSWORD=[password]  
DB_PORT=[database port]
```





To run project







```
npm start
```



Method	API Endpoint	Parameters Type	Query Parameters	Data	Description
GET	/api/recipes	-	?recipeld=int, ?userId=int, ?difficulty=string, ?start, ?count	-	Return recipes of said parameters
GET	/api/recipes	-	-	-	Return all created recipes from database
GET	/api/recipes/id	int	-	-	Return a recipe of said ID from database
PUT	/api/recipes/id	int	-	token	Updates recipe of said ID
PUT	/api/comments/id	int	-	token	Updates comment of said ID
DELETE	/api/recipes/id	int	-	token	Deletes recipe of said ID
DELETE	/api/comments/id	int	-	token	Deletes comment of said ID
POST	/api/recipes	-	-	token, new recipe	Create new recipe
POST	/api/comments	-	-	token, new comment, recipeld	Create new comment
POST	/api/favorite	-	-	token, recipeld	Add to favorites
POST	/api/like	-	-	token, recipeld	Like an recipe
POST	/auth/login	-	-	login,password	Log in
POST	/auth/register	-	-	login,password,password2,email	Create new user

Wybrane technologie:

Front-end Technology	Logo
TS	 TYPESCRIPT
Vue	 VUEJS
Pinia	
Tailwind	 TAILWINDCSS
Vite	 VITE
Vitest	
Cypress	 CYPRESS

Back-end Technology	Logo
TS	 TS-NODE
NodeJS	 NODE JS
PostgreSQL	 POSTGRESQL
Express	 EXPRESS JS
Postman	 POSTMAN
Vitest	
JWT	 JWT

Figma

Wizualizacja graficzna - Figma:



