Data storage and description of the main features of the program (functionality of the main scripts).

WWW *- root directory*

Files:

.htaccess - redirect all requests to the index.php script
index.php - main script, define routes and their handlers
.env - contain environment variables (PREFIX = "~khometym")
init.sh - init database files and manage access rights

Folders:

- ***common***
  - ***Database***
    Database.php - read/write database file, get with pagination, create and update
    Recipe.php - recipe class capable of validation (content field is vulnerable to XSS attacks!!!)
    database.json - contain recipes
  - ***Router***
    HTTPException.php - HTTP exception throwing functionality
    Request.php - class containing request information
    Response.php - class containing response methods
    Router.php - class containing methods for registering routes with handlers
    urlBuilder.php - function for building URL from sub-pathes
  - ***Validator***
    Validator.php - class containing validation methods
  - ***Authentication***
    Authentication.php - provide functionality of register/login and password hashing/verifying
    credentials.env - contain admin hashed password
- ***public***
  - ***assets***
    - ***favicon*** *- folder for containing project favicons*
    - ***uploads*** *- folder for user uploads*
  - ***docs*** *- static phpDoc documentation*
  - ***scripts***
    - ***common***

send-data.js - function sending requests and handling responses

login-page.js - script handling login form request

register-page.js - script handling register form request

operate-recipe-page.js - script handling create/update recipe requests

- ***styles*** *- folder containing project styles*

- ***views*** *- folder containing project pages and templates*
  - ***components*** *- folder containing reusable components*

Router module:

The Router module was inspired by Express.js, a Node.js backend library. It partially implements the Chain of Responsibility pattern.

- **Features**
  - **Dynamic routing (parameters in URL)**
  - **HTTP request parsing**
  - **Easy to use handlers system**

- **Example**

```php
$router = new Router();

// handling not found route request
$router->notFound(function (Request $request, Response $response) {
  $response->renderPage("404", ["uri" => $request->URI]);
});

// handling static GET request
$router->get("/", function (Request $request, Response $response) {
  $response->renderPage("home");
});

// handling dynamic GET request
$router->get("/recipes/:id", function (Request $request, Response $response) {
  $id = $request->parameters->id;

  echo $id;
});
```

Detailed documentation you can find at:
https://zwa.toad.cz/~khometym/public/docs