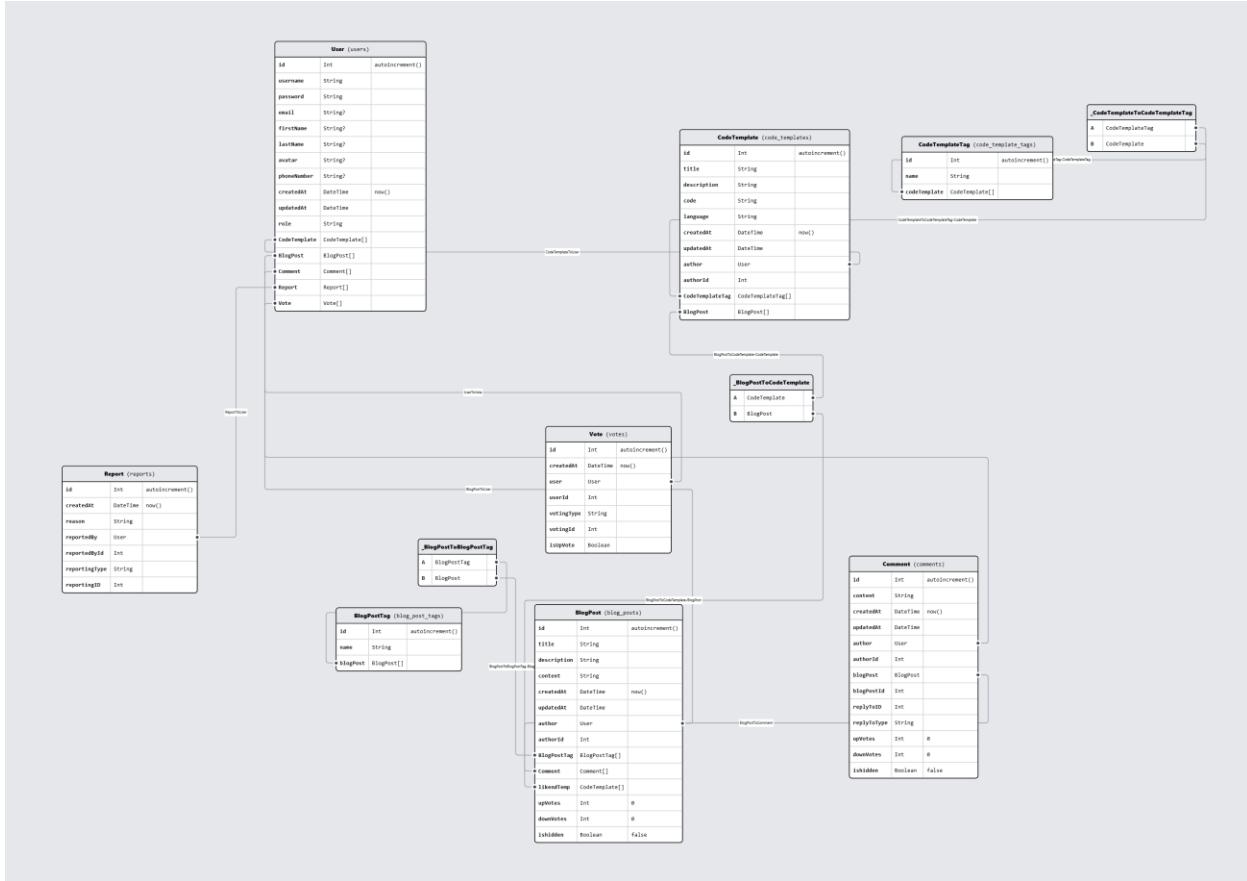


Project Documentation

```
// create summary of documentation here
```

Schema Design



Collection: api

Collection: {reportId}

End-point: Hide a reported comment/blog post by report ID

Method: PUT

`{baseUrl}/api/reports/:reportId`

This endpoint takes the reportId of the a specific report and returns the updated record (the blog_post or comment that was hidden)

Headers

Content-Type	Value
Accept	application/json

Body (raw)

```
{} // no body should be passed in
```

Response: 200 (record found)

```
{
  "updatedRecord": {
    "updatedRecord": {
      "id": 1,
      "title": "<string>",
      "description": "<string>",
      "content": "<string>",
      "createdAt": "2024-11-03T17:27:41.186Z",
      "updatedAt": "2024-11-03T22:22:36.356Z",
      "authorId": <integer>, // authorId of the person who owns the
      "upVotes": 0,
      "downVotes": 0,
      "ishidden": true
    }
  }
}
```

Response: 400

```
{ "error": "Report ID is required" }
```

Response: 403

```
{ "error": "Forbidden" }
```

Response: 404

```
{ "error": "Report not found" }
```

Response: 405

```
{ "error": "Method not allowed" }
```

Response: 500

```
{"error": "Error updating record", "message":error.message} //where
error.message is the specific message that failed
```

End-point: Get comments

Retrieves all comments as admin.

Method: GET

```
{{baseUrl}}/api/admin/comments?sortByReports=true
```

Query Params

Param	value
sortByReports	false

Authentication bearer

Param	value	Type
token	<code>{{admintoken}}</code>	string

Response: 200

Response: 500

End-point: Get blogs

Retrieves all blogs as admin.

Method: GET

`{{baseUrl}}/api/admin/blogs?sortByReports=true`

Query Params

Param	value
sortByReports	false

Authentication bearer

Param	value	Type
token	<code>{{admintoken}}</code>	string

Response: 200

Response: 500

End-point: Retrieve all reports

Method: GET

`{{baseUrl}}/api/admin/reports`

This endpoint returns a list of all reports sorted by creation date

Headers

Content-Type	Value
Accept	application/json

Body (raw)

{ } //no body

🔑 Authentication bearer

Param	value	Type
token	{{{admintoken}}}	string

Response: 200

```
[  
  {  
    "id": "<integer>",  
    "reportingType": "<string>",  
    "reportingID": "<integer>",  
    "createdAt": "<dateTime>",  
    "updatedAt": "<dateTime>"  
  },  
  {  
    "id": "<integer>",  
    "reportingType": "<string>",  
    "reportingID": "<integer>",  
    "createdAt": "<dateTime>",  
    "updatedAt": "<dateTime>"  
  }  
]
```

Response: 403

```
{ "error": "Forbidden" } //not admin
```

Response: 405

Response: 500

End-point: Create a new blog post

Method: POST

{}{{baseUrl}}/api/blogs

Headers

Content-Type	Value
Content-Type	application/json

Headers

Content-Type	Value
Accept	application/json

Body (raw)

```
{  
  "title": "<string>", //required  
  "description": "<string>", //required  
  "content": "<string>", //required  
  "BlogPostTag": ["<string>", "<string>"],  
  "likendTemp": [<integer>] //integer no quotation marks, api will not  
parseInt  
}
```

🔑 Authentication bearer

Param	value	Type
token	{token}	string

Response: 201

```
{  
  "id": "<integer>",  
  "title": "<string>",  
  "description": "<string>",  
  "content": "<string>",  
  "authorId": "<integer>",  
  "BlogPostTag": [  
    {  
      "id": "<integer>"  
    },  
    {  
      "id": "<integer>"  
    }  
  ],  
  "likendTemp": [  
    {  
      "id": "<integer>"  
    },  
    {  
      "id": "<integer>"  
    }  
  ]  
}
```

Response: 400

```
{ "error": "Missing required fields" }
```

Response: 404

```
{ "error": "User not found" }
```

Response: 500

```
{ "error": "Error updating database", "message": error.message }
```

End-point: Get all blog posts

Method: GET

```
{}{{baseUrl}}/api/blogs?search=<string>&sortBy=rating&page=1&limit=10
```

Headers

Content-Type	Value
Accept	application/json

Query Params

Param	value
search	
sortBy	rating
page	1
limit	10

Response: 200

```
[  
  {  
    "id": "<integer>",  
    "title": "<string>",  
    "description": "<string>",  
    "content": "<string>",  
    "authorId": "<integer>",  
    "BlogPostTag": [  
      {  
        "id": "<integer>"  
      },  
      {  
        "id": "<integer>"  
      }  
    ],  
    "likendTemp": [  
      {  
        "id": "<integer>"  
      },  
      {  
        "id": "<integer>"  
      }  
    ]  
  },  
  {
```

```
"id": "<integer>",
"title": "<string>",
"description": "<string>",
"content": "<string>",
"authorId": "<integer>",
"BlogPostTag": [
  {
    "id": "<integer>"
  },
  {
    "id": "<integer>"
  }
],
"likendTemp": [
  {
    "id": "<integer>"
  },
  {
    "id": "<integer>"
  }
]
}
```

Response: 500

 /api/blogs/[blogId]

 /api/blogs/[blogId]/[commentId]

End-point: Report a comment

Allows a user to report a comment for inappropriate content.

Method: POST

`{{baseUrl}}/api/blogs/:blogId/comments/:commentId/report`

Headers

Content-Type	Value
Content-Type	application/json

Body (raw)

```
{
  "reason": "<string>"
}
```

Response: 201

Response: 400

Response: 401

Response: 404

Response: 405

Response: 500

End-point: Retrieve a comment by ID

Retrieve a comment by its ID. If the comment is hidden, only the author or an admin can view it.

Method: GET

`{{baseUrl}}/api/blogs/:blogId/comments/:commentId`

Headers

Content-Type	Value
Accept	application/json

 **Authentication bearer**

Param	value	Type
token	<code>{{token}}</code>	string

Response: 200

```
{  
  "id": "<integer>",  
  "content": "<string>",  
  "authorId": "<integer>",  
  "blogPostId": "<integer>",  
  "ishidden": "<boolean>"  
}
```

Response: 400

Response: 403

Response: 404

Response: 500

End-point: Add a new comment

Adds a new comment to a blog post or another comment.

Method: POST

`{{baseUrl}}/api/blogs/:blogId/comments`

Headers

Content-Type	Value
Content-Type	application/json

Body (raw)

```
{  
  "content": "<string>",  
  "replyToID": "3",  
  "replyToType": "blog_post"  
}
```

Authentication bearer

Param	value	Type
token	<code>{{token}}</code>	string

Response: 201

Response: 400

Response: 404

Response: 500

End-point: Get comments

Retrieves comments for a specific blog post. Admins can view all comments, including hidden ones. Regular users can only view their own comments and non-hidden comments.

Method: GET

`{{baseUrl}}/api/blogs/:blogId/comments?page=1&limit=10&sortBy=createdAt`

Query Params

Param	value
page	1
limit	10
sortBy	createdAt

Authentication bearer

Param	value	Type
token	<code>{{token}}</code>	string

Response: 200

Response: 500

End-point: Report a blog post

Method: POST

`{{baseUrl}}/api/blogs/:blogId/report`

Headers

Content-Type	Value
Content-Type	application/json

Headers

Content-Type	Value
Accept	application/json

Body (raw)

```
{  
  "reason": "<string>"  
}
```

Response: 201

```
{  
  "report": {  
    "id": "<integer>",  
    "reportedById": "<integer>",  
    "reason": "<string>",  
    "reportingType": "<string>",  
    "reportingID": "<integer>",  
    "createdAt": "<dateTime>"  
  }  
}
```

Response: 400

Response: 403

Response: 404

Response: 405

Response: 500

End-point: Get a blog by ID

Retrieve a blog by its ID. If the blog is hidden, only the author or an admin can view it.

Method: GET

`{{baseUrl}}/api/blogs/:blogId`

Headers

Content-Type	Value
Accept	application/json

Body (raw)

Response: 200

```
{  
  "id": "<integer>",  
  "title": "<string>",  
  "description": "<string>",  
  "content": "<string>",
```

```
        "ishidden": "<boolean>",
        "authorId": "<integer>"
    }
```

Response: 403

Response: 404

Response: 500

End-point: Edit a blog by ID

Edit a blog by its ID. Only the author can edit the blog.

Method: PUT

`{baseUrl}/api/blogs/:blogId`

Headers

Content-Type	Value
Content-Type	application/json

Headers

Content-Type	Value
Accept	application/json

Body (raw)

```
{
    "title": "<string>",
    "description": "<string>",
    "content": "<string>",
    "BlogPostTag": ["<string>", "<string>"],
    "likendTemp": [
        1 //must be integer
    ]
}
```

🔑 Authentication bearer

Param	value	Type
token	<code>{token}</code>	string

Response: 200

```
{
    "id": "<integer>,"
```

```
"title": "<string>",
"description": "<string>",
"content": "<string>",
"BlogPostTag": ["<string>", "<string>"],
"likendTemp": ["<integer>", "<integer>"]
}
```

Response: 403

Response: 404

Response: 500

End-point: Delete a blog by ID

Delete a blog by its ID. Only the author can delete the blog.

Method: DELETE

`{{baseUrl}}/api/blogs/:blogId`

Response: 204

Response: 403

Response: 404

Response: 500

Collection: login

End-point: Login a user

Authenticate a user with username and password and return an access token.

Method: POST

`{{baseUrl}}/api/users/login`

Headers

Content-Type	Value
--------------	-------

Content-Type	Value
Content-Type	application/json

Headers

Content-Type	Value
Accept	application/json

Body (raw)

```
{  
  "username": "user",  
  "password": "password"  
}
```

Response: 200

```
{  
  "accessToken": "<string>"  
}
```

Response: 401

```
{  
  "error": "<string>"  
}
```

Response: 405

```
{  
  "error": "<string>"  
}
```

Response: 500

```
{  
  "error": "<string>"  
}
```

End-point: Login an admin

Authenticate a user with username and password and return an access token.

Method: POST

`{{baseUrl}}/api/users/login`

Headers

Content-Type	Value
Content-Type	application/json

Headers

Content-Type	Value
--------------	-------

Content-Type	Value
Accept	application/json

Body (raw)

```
{  
  "username": "admin",  
  "password": "admin_password"  
}
```

Response: 200

```
{  
  "accessToken": "<string>"  
}
```

Response: 401

```
{  
  "error": "<string>"  
}
```

Response: 405

```
{  
  "error": "<string>"  
}
```

Response: 500

```
{  
  "error": "<string>"  
}
```

Collection: profile

End-point: Get user profile

Retrieves the profile of the authenticated user.

Method: GET

`{{baseUrl}}/api/users/profile`

Headers

Content-Type	Value
Accept	application/json

Response: 200

```
{  
  "profile": {  
    "email": "<string>",  
    "firstName": "<string>",  
    "lastName": "<string>"  
  }  
}
```

```
        "lastName": "<string>",
        "avatar": "<string>",
        "username": "<string>",
        "role": "<string>"
    }
}
```

Response: 404

```
{
    "error": "<string>"
}
```

Response: 500

```
{
    "error": "<string>"
}
```

End-point: Update user profile

Updates the profile of the authenticated user.

Method: PUT

`{{baseUrl}}/api/users/profile`

Headers

Content-Type	Value
Accept	application/json

Body (raw)

```
{
    "email": "ltest@gmail.com",
    "firstName": "test",
    "lastName": "2dd",
    "avatar": "/output.png"
}
```

🔑 Authentication bearer

Param	value	Type
token	<code>{{token}}</code>	string

Response: 200

```
{
    "profile": {
        "email": "<string>",
        "firstName": "<string>",
        "lastName": "<string>",
        "avatar": "<string>"
```

```

        }
    }

Response: 400
{
    "error": "<string>"
}

Response: 404
{
    "error": "<string>"
}

Response: 405
{
    "error": "<string>"
}

Response: 500
{
    "error": "<string>"
}
-----
```

End-point: Upload profile picture file

Method: PUT

`{{baseUrl}}/api/users/uploadAvatar`

Headers

Content-Type	Value
Content-Type	multipart/form-data

Body formdata

Param	value	Type
file	/C:/Users/lilyx/OneDrive/Pictures/sem211styeartimetable.png	file

🔑 Authentication bearer

Param	value	Type
token	<code>{{token}}</code>	string

End-point: get photo

Method: GET

`{{baseUrl}}/api/users/getAvatar?id=2`

Query Params

Param	value
id	2

Collection: register

End-point: Register a new user

Registers a new user with a username and password. If the role is ADMIN, an adminSecret is required.

Method: POST

`{baseUrl}/api/users/register`

Headers

Content-Type	Value
Content-Type	application/json

Headers

Content-Type	Value
Accept	application/json

Body (raw)

```
{
  "username": "<string>",
  "password": "<string>",
  "role": "<string>",
  "adminSecret": "<string>"
}
```

Response: 201

```
{
  "user": {
    "id": "<string>",
    "username": "<string>",
    "role": "<string>"
  }
}
```

Response: 400

```
{
  "error": "<string>"
}
```

Response: 403

```
{  
  "error": "<string>"  
}
```

Response: 405

```
{  
  "error": "<string>"  
}
```

Response: 500

```
{  
  "error": "<string>"  
}
```

End-point: refresh accesstoken

Method: POST

`{{baseUrl}}/api/users/refresh`

Body (raw)

```
{ "refreshToken": "{{refreshToken}}" }
```

 **Authentication** noauth

Param	value	Type
-------	-------	------

End-point: refresh adminAccesstoken

Method: POST

`{{baseUrl}}/api/users/refresh`

Body (raw)

```
{ "refreshToken": "{{adminRefreshToken}}" }
```

 **Authentication** noauth

Param	value	Type
-------	-------	------

End-point: Create a vote

Allows a user to upvote or downvote a blog post or comment.

Method: POST

`{{baseUrl}}/api/vote`

Headers

Content-Type	Value
Content-Type	application/json

Body (raw)

```
{  
  "votingId": 1,  
  "votingType": "blog_post",  
  "isUpvote": true //no quotation marks around true/false values  
}
```

Authentication bearer

Param	value	Type
token	{{token}}	string

Response: 200

Response: 400

Response: 404

Response: 500

End-point: Delete a vote

Allows a user to delete their vote on a blog post or comment.

Method: DELETE

`{{baseUrl}}/api/vote`

Headers

Content-Type	Value
Content-Type	application/json

Body (raw)

```
{  
  "votingId": 1,  
  "votingType": "blog_post"  
}
```

 **Authentication bearer**

Param	value	Type
token	{{token}}	string

Response: 200

Response: 400

Response: 404

Response: 500

 **Collection: templates**

End-point: visitor: get all or filtered templates

Method: GET

`{{baseUrl}}/api/code/templates`

Body (raw)

Query Params

Param	value
title	
tags	[]
code	hello
page	
pageSize	

 **Authentication noauth**

Param	value	Type
	null	

Response: undefined

null

End-point: specific user: get all or filtered

Method: GET

`{{baseUrl}}/api/code/templates?authorId=2`

Query Params

Param	value
title	
code	
tags	[strings,string2]
authorId	2

🔑 Authentication bearer

Param	value	Type
token	<code>{{token}}</code>	string

Response: undefined

`null`

End-point: add new code template (authenticated user)

Method: POST

`{{baseUrl}}/api/code/templates/add/`

Body (raw)

```
{  
    //example of a body to send to the user: code, language and title are  
    required  
    "title": "this file",  
    "code": "System.out.println()",  
    "language": "java",  
    "description": "print hello world",  
    "tags": ["tag1", "tag2"]  
}
```

🔑 Authentication jwt

Param	value	Type
-------	-------	------

Response: undefined

`null`

End-point: get specific code template by id

Method: GET

`{{baseUrl}}/api/code/templates/:id`

Response: undefined

`null`

End-point: edit specific code template (authenticated user)

Method: PUT

`{{baseUrl}}/api/code/templates/:id`

Body (raw)

```
{  
  //example body none of these fields are necessary, defaults to whatever the  
  //code template was before the change  
  "title": "this file",  
  "code": "print('hello world')",  
  "language": "python",  
  "description": "print hello world",  
  "tags": ["tag1", "other tag", "tag4"]  
}
```

🔑 Authentication bearer

Param	value	Type
token	<code>{{token}}</code>	string

Response: undefined

`null`

End-point: delete specific code template by id

Method: DELETE

`{{baseUrl}}/api/code/templates/:id`

Body (raw)

🔑 Authentication jwt

Param	value	Type

End-point: execute code

Method: POST

`{{baseUrl}}/api/code/execute`

Body (raw)

```
{  
  //example body code for python  
  "code": "print(\"hello world\")",  
  "language": "python"  
}
```

Response: undefined

`null`

Powered By: [postman-to-markdown](#)