

API DOCUMENTATION

(v1.0.0)

Author/s:

Anecito, Mary Kate F.

Flores, Marvin Edilson C.

Paulino, Juan Miguel P.

Ramos, King Emmanuel S.

Tomate, Alexene Faith S.

Developer/s:

Anecito, Mary Kate F.

Flores, Marvin Edilson C.

Paulino, Juan Miguel P.

Ramos, King Emmanuel S.

Tomate, Alexene Faith S.

Table of Contents

<u>A. Account Module</u>

1. Registration / Sign Up

2.Verify Email

3.Login

4. Change Password

5. Forgot Password

6. Delete Account

7.Request Forgot Password OTP

8.Request Change Password OTP

9.Change Password Verify OTP

10.Forgot Password Verify OTP

<u>B. Studysset Module</u>

1.Create Study set

2.View Study set

3.Update Study set

4.Delete Study set

<u>C.Flash Card Module</u>

1.Create Flashcard

2.View Flashcard

3.Update Flashcard

4.Delete Flashcard

<u>D. Test Module</u>

1.Randomize Flashcard

2.Validate Learner Answer

3.Save Results

4.Save Batch

<u>E. Report Module</u>

1.Save Test Scores

2.View Test Scores

<u>F. Contact Inquiry Module</u>

1.Create Inquiry

<u>G. FAQ Module</u>

1.Create FAQ

2.View FAQ

3.Update FAQ

4. Delete FAQ

Account Module

This module includes API functionalities for managing accounts in the database.

1. Registration / Sign Up

- This API allows creation or addition of accounts in the database.

Endpoint: `http://localhost:8000/account/register/`

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"email"	Refers to the learner's email.	String	Mandatory	"test@email.com"
"password"	Refers to the learner's password.	String	Mandatory	"Test1234!"
"password 2"	Refers to the learner's password again to verify if they have typed it properly	String	Mandatory	"Test1234!"

Sample Request Body: (json)

```
{  
  "email": "printer@gmail.com",  
  "password": "inkjectPrinter1!",  
  "password2": "inkjectPrinter1!"  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"data"		String	

"message"	Indicates the actual description of the response.	String	"User created successfully"
-----------	---	--------	-----------------------------

Sample Response Body: (json)

a. Response Status: 201 (Created)

- Request to create a learner in the database is successfully completed.

```
{
  "successful": true,
  "message": "User created successfully"
}
```

b. Response Status: 404 (Not Found)

- Request body has missing mandatory fields.

```
{
  "successful": false,
  "message": "Product name or product quantity is not defined."
}
```

c. Response Status: 500 (Internal Server Error)

- Error in server or database.

```
{
  "successful": false,
  "message": "Database table not found."
}
```

d. Response Status: 400 (Bad Request)

- User did not put the correct requirements for the fields

```
{
  "email": [
    "Enter a valid email address."
  ],
  "password": [
    "Ensure this field has at least 8 characters."
  ],
  "password2": [
    "Ensure this field has at least 8 characters."
  ]
}
```

2. Verify Email

- This API allows learners to verify their accounts in order to access athAlna.

Endpoint: `http://localhost:8000/account/verify-email/`

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"otp"	Refers to the learner's otp that they have received through their email.	String	Mandatory	"123456"

Sample Request Body: (json)

```
{  
  "otp": "168346",  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Successfully added a product"

Sample Response Body: (json)

e. Response Status: 200 (OK)

- Request to verify a learner's email is successful.

```
{  
  "successful": true,  
  "message": "Email verified successfully"  
}
```


f. Response Status: 200 (OK)

- Request to verify denied because learner's account is already verified.

```
{
  "successful": false,
  "message": "Email already verified"
}
```

g. Response Status: 400 (Bad Request)

- The learner's input has an error .

```
{
}
```

h. Response Status: 400 (Bad Request)

- OTP is no longer in the database.

```
{
  "successful": False,
  "message": "Invalid OTP"
}
```

3. Login

- This API allows learners to log in to their accounts.

Endpoint: <http://localhost:8000/account/login/>

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"email"	Refers to the learner's otp that they have received through their email.	String	Mandatory	"123456"
"password"	Refers to the learner's password.	String	Mandatory	"Password1!"

Sample Request Body: (json)

```
{
  "email" : "testuser8008@gmail.com",
  "password" : "testest1"
```

}

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Successfully added a product"

Sample Response Body: (json)

i. Response Status: 200 (OK)

- Request to verify a learner's email is successful.

```
{
  "successful": true,
  "message": "Email verified successfully"
}
```

j. Response Status: 200 (OK)

- Request to verify denied because learner's account is already verified.

```
{
  "successful": false,
  "message": "Email already verified"
}
```

k. Response Status: 400 (Bad Request)

- The learner's input has an error .

```
{
  "successful": false,
  "message": "Invalid OTP"
}
```

l. Response Status: 400 (Bad Request)

- The OTP currently stored in the user's account is invalid.

```
{
  "Error": "Invalid Error"
}
```

m. Response Status: 401 (Unauthorized)

- Product name to be added already exists in the database.

```
{  
    "successful": false,  
    "message": "Product Name already exists."  
}
```

4. Forgot Password

- This API allows learners to change their password if they can't remember their password.

Endpoint: <http://localhost:8000/account/forgot-password/<uidb64>/<token>>

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"password"	Refers to the learner's new password	String	Mandatory	"Test3214!"
"confirm_password"	Refers to the confirmation of their new password	String	Mandatory	"Test3214!"
"uuid"	Refers the users UUID needed for the changing of passwords	String	Mandatory	"A8"
"token"	Refers to the user's token given to them in order for them to validate their accounts for password change	String	Mandatory	"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiOilxMjM0NTY3ODkwIiwiaWF0IjE6bmFtZSI6IkpvaG4gRG9Iliwia

				WF0Ijox NTE2Mj M5MDIyf Q.S0meR anDomC haracters Here”
--	--	--	--	---

Sample Request Body: (json)

```
{
  "password": "Test1234!",
  "confirm_password": "Test1234!",
  "Uuid": "A8",
  "Token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMONTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.S0meRanDomCharactersHere"
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Password updated successfully "
"error"	Indicates the password reset link	String	"http://localhost/uuid/token/"

Sample Response Body: (json)

n. Response Status: 400 (Bad Request)

- Request to verify a learner's forgot password token has expired.

```
{
  "message": "Token is not valid, please request a new one",
  "successful": False
}
```

o. Response Status: 200 (OK)

- Request to verify success and their password has been changed.

```
{
    "message": "Password Reset Successfully",
    "successful": True
}
```

p. Response Status: 400 (Bad Request)

- Athena sent itself the wrong token due to human error .

```
{
    "message": "Invalid token",
    "successful": False
}
```

q. Response Status: 400 (Bad Request)

- An exception has occurred.

```
{
    "error": str(e)
}
```

5. Change Password

- This API allows learners to change their password if they remember their old password..

Endpoint: <http://localhost:8000/account/change-password/<uidb64>/<token>>

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"Old password"	Refers to the learner's new password	String	Mandatory	"Test321 4!"
"confirm_p assword"	Refers to the confirmation of their new password	String	Mandatory	"Test321 4!"

"uuid"	Refers the users UUID needed for the changing of passwords	String	Mandatory	"A8"
"token"	Refers to the user's token given to them in order for them to validate their accounts for password change	String	Mandatory	"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMONTY3ODkwliwiOiJpvaG4gRG9IiwiaWF0IjoxNTE2MjM5MDIyfQ.S0meRanDomCharactersHere"

Sample Request Body: (json)

```
{
  "password": "Test1234!",
  "confirm_password": "Test1234!",
  "Uuid": "A8",
  "Token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMONTY3ODkwliwiOiJpvaG4gRG9IiwiaWF0IjoxNTE2MjM5MDIyfQ.S0meRanDomCharactersHere"
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true

"message"	Indicates the actual description of the response.	String	"Password updated successfully "
"error"	Indicates the password reset link	String	"http://localhost/uuid/to ken/"

Sample Response Body: (json)

r. Response Status: 400 (Bad Request)

- Request to verify a learner's forgot password token has expired.

```
{
  "message": "Token is not valid, please request a new one",
  "successful": False
}
```

s. Response Status: 200 (OK)

- Request to verify success and their password has been changed.

```
{
  "message": "Password updated successfully",
  "successful": True
}
```

t. Response Status: 400 (Bad Request)

- Athena sent itself the wrong token due to human error .

```
{
  "message": "Invalid token",
  "successful": False
}
```

u. Response Status: 400 (Bad Request)

- An exception has occurred.

```
{
  "error": str(e)
}
```

6. Delete Account

- This API allows learners to delete their account as they wish.

Endpoint: http://localhost:8000/account/delete-account/

HTTP Method: DELETE

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
message	Description of the operation result	string	Mandatory	"User deleted successfully"
successful	Indicates whether the request was successful	boolean	Mandatory	true

Sample Request Body: (json)

```
{  
  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Successfully added a product"
"error"	Indicates the password reset link	String	"http://localhost/uuid/token/"

Sample Response Body: (json)

v. **Response Status:** 401 (Unauthorized)
- Authentication headers not provided

```
{  
  
}
```

7. Request Forgot Password OTP

- This API allows users to request to reset their password by sending an OTP (One-Time Password) to their email for verification.

Endpoint: http://localhost:8000/account/change-password-request/

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
email	User's registered email address	String	Mandatory	"user@example.com"

Sample Request Body: (json)

```
{  
  "email" : "user@example.com"  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"An OTP has been sent to your email to verify your identity"

Sample Response Body: (json)

w. Response Status: 200(OK)

- Authentication headers not .provided

```
{  
  "message": "An OTP has been sent to your email to verify your identity",  
  "successful": true  
}
```

8. Request Change Password OTP

- This API allows users to request to change their password by sending an OTP (One-Time Password) to their email for verification.

Endpoint: http://localhost:8000/account/password-chang-request/

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
email	User's registered email address	String	Mandatory	"user@example.com"

Sample Request Body: (json)

```
{  
  "email" : "user@example.com"  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"An OTP has been sent to your email to verify your identity"

Sample Response Body: (json)

x. Response Status: 200(OK)

- Authentication headers not provided

```
{  
  "message": "An OTP has been sent to your email to verify your identity",  
  "successful": true  
}
```

9. Verify Forgot Password OTP

- This API allows learners to input the OTP code they have received

Endpoint: http://localhost:8000/account/verify-forgot-password-otp

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"otp"	Refers to the learner's otp that they have received through their email.	String	Mandatory	"123456"

Sample Request Body: (json)

```
{  
  "otp": "168346",  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Successfully added a product"
"password_reset_link"	Indicates the password reset link	String	"http://localhost/uuid/token/"

Sample Response Body: (json)

y. Response Status: 200 (OK)

- Request to verify a learner's forgot password otp is successful.

```
{  
  "message": "OTP verified successfully",  
  "successful": True,  
  "password_reset_link": abslink  
}
```

z. Response Status: 200 (OK)

- Request to verify denied because learner's account is already verified.

```
{
  "successful": false,
  "message": "Email already verified"
}
```

aa. Response Status: 400 (Bad Request)

- The learner's input has an error .

```
{
}
```

bb. Response Status: 400 (Bad Request)

- OTP is no longer in the database.

```
{
  "message": "Invalid or expired OTP",
  "successful": False
}
```

10. Verify Change Password OTP

- This API allows learners to input the OTP code they have received

Endpoint: <http://localhost:8000/account/verify-forgot-password-otp>

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"otp"	Refers to the learner's otp that they have received through their email.	String	Mandatory	"123456"

Sample Request Body: (json)

```
{
  "otpl": "168346",
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Successfully added a product"
"password_reset_link"	Indicates the password reset link	String	"http://localhost/uuid/token/"

Sample Response Body: (json)

cc. Response Status: 200 (OK)

- Request to verify a learner's forgot password otp is successful.

```
{
  "message": "OTP verified successfully",
  "successful": True,
  "password_reset_link": abslink
}
```

dd. Response Status: 200 (OK)

- Request to verify denied because learner's account is already verified.

```
{
  "successful": false,
  "message": "Email already verified"
}
```

ee. Response Status: 400 (Bad Request)

- The learner's input has an error .

```
{
}
```

ff. Response Status: 400 (Bad Request)

- OTP is no longer in the database.

```
{
  "message": "Invalid or expired OTP",
  "successful": False
}
```

}

2. Studyset Module

This module includes API functionalities for managing StudySets in the database.

1.Create Studyset

- This API allows creation or addition of StudySets in their database.

Endpoint: <http://localhost:8000/studyset/save/>

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"learner_instance"	Refers to the learner's instance ID	int	Mandatory	"1"
"title"	Study set field requirement	String	Mandatory	"Networking"
"description"	Study set field requirement	String	Mandatory	"This is all about networking"
"subjects"	Study set field requirement from the drop down choices	String	Mandatory	"TECH"

Sample Request Body: (json)

```
{  
  "Learner_instance": "1",  
  "title": "Networking",  
  "description": "This is all about networking"  
  "subject": "TECH"  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Successfully added a product"

Sample Response Body: (json)

gg. Response Status: 200 (OK) : HTTP_201_CREATED

```
{
  "successful": true,
  "message": "StudySet created successfully"
  "data": serializer.data,
}
```

hh. Response Status: 404 (Not Found) : HTTP_400_BAD_REQUEST

```
{
  "successful": False
  "message": "Study Set could not be created, please try again.",
  "Errors" : serializer.errors,
}
```

2.View Studysset

- This API allows Learners to view the StudySet in their database.

Endpoint: `http://localhost:8000/studysset/`

HTTP Method: GET

Request Body: N/A

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Successfully added a product"
"data"	Contains the information of each product: prod_id – ID of the product prod_name – product name prod_desc – actual description of the product prod_qty – remaining quantity of the product	Array of Objects	{ "prod_id": 1, "prod_name": "printer", "prod_desc": "inkjet printer", "prod_qty": 4 }

Sample Response Body: (json)

a. Response Status: 200 (OK)

- Request to add a product in the database is successfully completed.

```
{
  "successful": true,
  "message": "Successfully got all products.",
  "data": [
    {
      "prod_id": 1,
      "prod_name": "printer",
      "prod_desc": "inkjet printer",
      "prod_qty": 4
    }
  ]
}
```

b. Response Status: 500 (Internal Server Error)

- Error in server or database.

```
{
  "successful": false,
  "message": "."
}
```

3. Update Studyset

- This API allows changing of the Studysets information.

Endpoint: <http://localhost:8000/studyset/update/<int:pk>>

HTTP Method: PUT

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"title"	Specifies the new quantities updated name	string	Mandatory	true
"description"	Specifies the new quantities updated description	string	Mandatory	true
"subjects"	Specifies the new quantities updated subjets	string	Mandatory	true

Sample Request Body: (json)

```
{
  "title": "",
  "description": "",
  "subjects": ""
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Study Set updated successfully"

Sample Response Body: (json)

a. Response Status: 200 (OK) : HTTP_200_OK

```
{
  "message": "Study Set updated successfully",
  "data": serializer.data,,
  "successful": true,
}
```

b. Response Status: 400 (Bad Request) : HTTP_400_BAD_REQUEST

```
{
  "message": "Study Set could not be updated, please try again",
  "errors": serializer.errors,
```

```
        "successful": false,  
    }  
}
```

4. Delete Study Set

- This API allows removal of studyset in the database.

Endpoint: http://localhost:8000/studyset/delete/<ID>/

HTTP Method: DELETE

Request Body: N/A

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Successfully deleted the product"
"data"	Indicates the actual data .	String	

Sample Response Body: (json)

a. Response Status: 200 (OK) : HTTP_200_OK

- Request to delete the product in the database is successfully completed.

```
{  
    "message": true,  
    "successful": "Successfully deleted the Studysset",  
    "data" : serializer.data  
}
```

c. Response Status: 400 (Bad Request) : HTTP_400_BAD_REQUEST

```
{
  "message": "Study set could not be deleted, please try again.",
  "successful": False,
  "errors": serializer.errors
}
```

3. Flashcard Module

This module includes API functionalities for managing StudySets in the database.

1. Save Flashcard

- This API allows creation or addition of flashcards in their respective studyset.

Endpoint: http://localhost:8000/flashcard/save/

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"question"	Flashcard field requirement	string	Mandatory	"What is the capital of France?"
"answer"	Flashcard field requirement	String	Mandatory	"Paris"
"Image"	Flashcard field requirement	jpg	Optional	null
"studyset_instance"	Refers to the learner's instance ID	int	Mandatory	"1"

Sample Request Body: (json)

```
{
  "question": "What is the capital of France?",
  "answer": "Paris",
  "image": null,
```

```
    "studysset_instance": "1",  
  }  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Flashcard created successfully"

Sample Response Body: (json)

ii. Response Status: 201 (OK) : HTTP_201_CREATED

- Request to add a flashcard in the studysset is successfully completed.

```
{  
    "successful": true,  
    "message": "Flashcard created successfully",  
    "data": serializer.data,  
    "Status" : HTTP_201_CREATED  
}
```

jj. Response Status: 400 (Not Found) : HTTP_400_BAD_REQUEST

- Request body has missing mandatory fields.

```
{  
    "successful": False,  
    "message": "Flashcard could not be created, please try again.",  
    "Errors" : serializer.errors,  
    "Status" : HTTP_400_BAD_REQUEST,
```

}

2. View All Flashcards

- This API allows viewing of information regarding all flashcards in the learner's study set.

Endpoint: <http://localhost:8000/flashcard/library/>

HTTP Method: GET

Request Body: N/A

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully retrieved or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Flashcard retrieved successfully t"
"data"	Contains the information of each product: "question": "", "answer": "", "image": , "studysset_instance": , "created_at": "2023-10-01T12:34:56Z",	Array of Objects	{ "question": "What is the capital of France?", "answer": "Paris", "image": null, "studysset_instance": 1, "created_at": "2023-10-01T12:34:56Z",

			}
--	--	--	---

Sample Response Body: (json)

a. Response Status: 200 (OK)

- Request to display existing flashcards in the database is successfully completed.

```
{
    response = self.get_paginated_response(serializer.data),
    response.status_code= HTTP_200_OK
    return response ,
    "data": [
        {
            "question": "What is the capital of France?",
            "answer": "Paris",
            "image": null,
            "studysset_instance": 1,
        }
    ]
}
```

b. Response Status: 404

- Error in server or database.

```
{
    "successful": false,
    "message": "No Flashcard found."
    "status": "HTTP_200_OK"
}
```

3.Update Flashcard

- This API allows updating existing flashcards.

Endpoint: http://localhost:8000/flashcard/update/<int:id>

HTTP Method: PUT

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"question"	Flashcard field requirement	string	Mandatory	"What is the capital of the philippines?"
"answer"	Flashcard field requirement	string	Mandatory	"Manila"
"image"	Flashcard field requirement		optional	null
"Studyset_instance "	Flashcard placement on the appointed studyset	Int	Mandatory	"1"

Sample Request Body: (json)

```
{
  "question": "What is the capital of the Philippines?",
  "answer": "Manila",
  "Image" : null,
  "studyset_instance": 1
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true

"message"	Indicates the actual description of the response.	String	"Flashcard updated successfully product"
"data"	"question": "", "answer": "", "image": , "studysset_instance": , "created_at": "2023-10-01T12:34:56Z", "updated_at": "2023-10-01T12:34:56Z",	Array of Objects	"question": "What is the capital of the Philippines?", "answer": "Manila", "Image" : null, "studysset_instance": 1 "created_at": "2023-10-01T12:34:56Z", "updated_at": "2023-10-01T12:34:56Z",
"status"		integer	HTTP_200_OK

Sample Response Body: (json)

a. Response Status: 200 (OK)

- Request to update the product's quantity in the database is successfully completed.

```
{
  "successful": true,
  "message" : "Flashcard updated successfully.",
  "data": serializer.data,
  "status": HTTP_200_OK
}
```

b. Response Status: 400 (Bad Request)

- Product does not exist in the database.

```
{
  "successful": false,
  "message" : "Flashcard could not be updated, please try again.",
  "errors": serializer.errors,
  "status": HTTP_400_BAD_REQUEST
}
```

4.Delete Flashcard

- This API allows removal of flashcards in their respective study sets.

Endpoint: http://localhost:8000/flashcard/delete/<int:id>

HTTP Method: DELETE

Request Body: N/A

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Flashcard deleted successfully."
"Data"	"question": "", "answer": "", "image": , "studysset_instance": , "created_at": "2023-10-01T12:34:56Z", "updated_at": "2023-10-01T12:34:56Z",	Array of Objects	"question": "What is the capital of the Philippines?", "answer": "Manila", "Image" : null, "studysset_instance": 1 "created_at": "2023-10-01T12:34:56Z", "updated_at": "2023-10-01T12:34:56Z",
"status"		integer	HTTP_200_OK

Sample Response Body: (json)

a. Response Status: 200 (OK)

- Request to delete the product in the database is successfully completed.

```
{  
  "successful": true,  
  "message" : "Flashcard deleted successfully.",  
  "data": serializer.data,  
  "status": HTTP_200_OK  
}
```

b. Response Status: 400 (Bad Request)

- Product does not exist in the database.

```
{
  "successful": false,
  "message" : "Flashcard could not be deleted , please try again.",
  "errors": serializer.errors,
  "status": HTTP_400_BAD_REQUEST
}
```

Test Module

This module includes API functionalities for managing Test Mode

1. Randomize Flashcards.

- This API Randomizes flashcards in their respective study sets upon selecting how many flashcards to use when using Test Mode..

Endpoint: <http://localhost:8000/test/randomize/>

HTTP Method: GET

Request Body:(json)

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates if the request was successful	Boolean	True
"message"	Message describing the result of the request	String	"Flashcards generated successfully."
"flashcard_ids"	List of randomized flashcard IDs	Array	[1, 2, 3]

Sample Response Body: (json)

c. Response Status: 200 (OK)

- Request to randomize is successfully completed.

```
{
  "successful": true,
  "message" : "Flashcards generated successfully.",
}
```

```

    "Flashcard_ids": [1,2,3],
    "status": HTTP_200_OK
}

```

d. Response Status: 400 (Bad Request)

- Flashcards requested to randomize exceeds flashcards that are available

```

{
    "successful": false,
    "message" : "The Number of questions exceeds the available flashcards.",,
    "status": HTTP_400_BAD_REQUEST
}

```

e. Response Status: 400 (Bad Request)

- Flashcards requested to randomize did not included number of flashcards

```

{
    "successful": false,
    "message" : "Please provide a valid number of questions.",,
    "status": HTTP_400_BAD_REQUEST
}

```

f. Response Status: 400 (Bad Request)

- Flashcards requested to randomize did not have any flashcard available

```

{
    "successful": false,
    "message" : "No Flashcard Available.",,
    "status": HTTP_400_BAD_REQUEST
}

```

2. Validate Learner's Answer API

- This API allows validation of learner's answers using AI.

Endpoint: http://localhost:8000/test/validate_learner_answer/<int:id>/

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
?learner_answer	Parameter that sends the learner answer to be validated		mandatory	?learner_answer=answer

Sample Request Body: (json)

```
{
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"message"	Indicates the answer that the user is trying to validate	String	Wireless fidelity
"is_correct"	Indicates the response of the AI validation.	Boolean	true
"successful"	Indicates if the operation is successful or not	Boolean	true

Sample Response Body: (json)

kk. Response Status: 200 (OK) : HTTP_200_OK

- Request to add a flashcard in the studyset is successfully completed.

```
{
  "message": "Validating learner answer with AI.",
  "is_correct": true,
  "Successful": true
}
```

ll. Response Status: 400 (Not Found) : HTTP_400_BAD_REQUEST

- Request body has missing mandatory fields.

```
{
  "message": "Flashcard not found.",
  "successful": False,
}
```

3. Save Test Results

- This API allows saves the test results of the learners.

Endpoint: http://localhost:8000/test/save/

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
?learner_answer	Parameter that sends the learner answer to be validated		mandatory	?learner_answer=answer

Sample Request Body: (json)

Response Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"message"	Indicates the status of the operation	String	mandatory	Test score saved successfully.
"data"	Indicates the data group	Boolean	mandatory	
"id"	Indicates if the operation is successful or not	Boolean	mandatory	5

"studysset_instance"	Indicates the id of the study set instance of the test	Integer	mandatory	2
"batch"	Indicates the batch id of the test	Integer	mandatory	5
"score"	Indicates the score the learner achieved	Integer	mandatory	1
"number_of_questions"	Indicates the number of questions the learner's test contains	Integer	mandatory	1
"Submitted_at"	Indicate the time the learner submitted the test	Datetime	mandatory	2025-03-28T12:03:19.244000Z
"Successful"	Indicates if the test has been recorded	String	mandatory	true

Sample Response Body: (json)

mm. Response Status: 201 (OK) : HTTP_201_CREATED

- Test information has been saved to the database.

```
{
  "message": "Test score saved successfully.",
  "Data": {
    "Id": 5,
    "Studysset_instance": 2,
    "Batch": 5,
    "Score": 1,
    "Number_of_questions": 1,
    "submitted_at": "2025-03-28T12:03:19.244000Z",
    "Successful": true
  }
}
```

nn. Response Status: 400 (Not Found) : HTTP_400_BAD_REQUEST

- Request body has missing mandatory fields.

```
{
  "message": "An error occurred while saving test results.",
  "error": str(e),
  "successful": False
}
```

4. Save Batch Information

- This API allows saves the test results of the learners.

Endpoint: http://localhost:8000/test/save/

HTTP Method: POST

Request Body: (json)

```
{  
  "Batch_id": "4f1cbd6a-820f-4148-9aec-3fe4153df665"  
}
```

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
?Batch_id	The batch id of the test	UUID	mandatory	?learner_answer=answer

Sample Request Body: (json)

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"message"	Indicates the status of the operation	String	Test batch created successfully.
"data"	Indicates the data group	Boolean	
"id"	Indicates if the operation is successful or not	Boolean	6

"batch_id"	Indicates the batch id of the test	UUID	4f1cbd6a-820f-4148-9aec-3fe4153df665
"created_at"	Indicate the time the learner submitted the test	Datetime	2025-03-28T12:03:19.244000Z
"Successful"	Indicates if the test has been recorded	String	true

Sample Response Body: (json)

oo. Response Status: 201 (OK) : HTTP_201_CREATED

- Test batch has been saved to the database.

```
{
  "message": "Test Batch created successfully.",
  "Data": {
    "Id": 6,
    "Batch_id": "4f1cbd6a-820f-4148-9aec-3fe4153df665",
    "created_at": "2025-03-28T12:52:44.536335Z",
    "successful": true
  }
}
```

pp. Response Status: 400 (Not Found) : HTTP_400_BAD_REQUEST

- Request body has missing mandatory fields.

```
{
  "message": "Test Batch could not be created, please try again.",
  "errors": {
    "batch_id": ["Invalid UUID format"]
  },
  "successful": false
}
```

Report Module

This module includes API functionalities for managing test results in the database.

1. Save Test Scores

- This API saves test scores of the learner to the database

Endpoint: <http://localhost:8000/report/save/>

HTTP Method: POST

Request Body: (json)

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not	boolean	True
"message"	Indicates the actual description of the response	string	"Test scores saved successfully."
"data"	An object containing the test score	array	"id": 1, "studysset_instance": 1, "score": 85, "number_of_questions": 10, "submitted_at": "2023-10-01T12:34:56Z"
"status"	HTTP status code	integer	HTTP_200_OK

Sample Response Body: (json)

a. Response Status: 200 (OK)

- Request to view test result in the database is successfully completed.

```
{  
  "successful": true,  
  "message": "Test scores found.",  
  "data":  
    {  
      "id": 1,  
      "studysset_instance": 1,
```

```

        "score": 7,
        "number_of_questions": 10,
        "submitted_at": "2023-10-01T12:34:56Z"
    }
},
    "status": 200
}

```

b. Response Status: 400

- Error in server or database.

```

{
    "successful": false,
    "message": "Test score could not be saved, please try again",
    "status": 400
}

```

2. View Test Scores

- This API gets the test scores by subject and date

Endpoint: <http://localhost:8000/report/list/>

HTTP Method: POST

Request Body: (json)

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not	boolean	True
"message"	Indicates the actual description of the response	string	"Test scores saved successfully."
"data"	Array of objects containing test result details	array	"id": 1, "studysset_instance": 1, "score": 85, "number_of_questions": 10,

			"submitted_at": "2023-10-01T12:34:56Z"
"status"	HTTP status code	integer	HTTP_200_OK

Sample Response Body: (json)

b. Response Status: 200 (OK)

- Request to view test result in the database is successfully completed.

```
{
  "successful": true,
  "message": "Test scores found.",
  "data": [
    {
      "id": 1,
      "studysset_instance": 1,
      "score": 7,
      "number_of_questions": 10,
      "submitted_at": "2023-10-01T12:34:56Z"
    }
  ],
  "status": 200
}
```

b.Response Status: 400

- Error in server or database.

```
{
  "successful": false,
  "message": "No test scores found.",
  "status": 400
}
```

Contact Inquiry Module

This module includes API functionalities for managing Learner’s messages of concerns or inquiries and stored in the database.

Contact Inquiry Create

- This API allows creation or addition of Inquiries in the database.

Endpoint: <http://localhost:8000/contact/>

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"name"	Refers to the learner's name.	String	Mandatory	"John Doe"
"email"	Indicates email of learner, to keep track and communicate to that learner	String	Optional	"John@test.com"
"message"	Indicates learners, message of his/her inquiry that will limit to 300 characters	string	Mandatory	"Dreams are the seeds of our future. Every idea you cultivate, every goal you strive for, they all add up to create the life you envision. Embrace your journey, with its ups and downs, for it's the challenges that shape us. Keep pushing forward, stay curious, and let passion guide you."

Sample Request Body: (json)

```
{  
  "name": "John Doe",  
  "email": "Test@test.com",  
  "message": "Dreams are the seeds of our future. Every idea you cultivate, every  
goal you strive for, they all add up to create the life you envision. Embrace your  
journey, with its ups and downs, for it's the challenges that shape us. Keep pushing  
forward, stay curious, and let passion guide you."  
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"Your contact inquiry has been submitted"
"data"	"name": , "email": , "message":,	Array of Objects	{ "name": "John Doe", "email": "Test@test.com", "message": "Dreams are the seeds of our future. Every idea you cultivate, every goal you strive for, they all add up to create the life you envision. Embrace your journey, with its ups and downs, for it's the challenges that shape us. Keep pushing forward, stay curious, and let passion guide you." }

Sample Response Body: (json)

a. Response Status: 200 (OK)

```
{  
  "message": "Your contact inquiry has been submitted.",  
  "data": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "john@example.com",  
    "message": "I have a question about the flashcard feature.",  
    "created_at": "2023-10-01T12:34:56Z"  
  }  
}
```

b. Response Status: 400 (Bad Request)

```
{
  "message": "Your contact inquiry could not be submitted, please try again.",
  "errors": {
    "name": ["Your name is needed so we know who to address."],
    "email": ["Please provide a valid email address."],
    "message": ["Your message is needed so we know how to help you."]
  }
}
```

FAQ Module

This module includes API functionalities for managing developers messages to their users (learner) about common problems faced when using the site and will be stored in the database.

1. FAQ Create

- This API allows creation or addition of Frequently Asked Questions(FAQs) in the database.

Endpoint: <http://localhost:8000/faq/save/>

HTTP Method: POST

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"question"	Refers to the developer inputting common	String	Mandatory	"How can I create flashcards on athAlna?"

	problems encountered when using the site			
"answer"	Indicates the solution paired with the problem.	String	Mandatory	<p>"You can create flashcards in two ways:</p> <p>Manually. Fill out the flashcard fields by entering questions and answers yourself.</p> <p>Automatically. Upload your notes, and athAlna will convert them into flashcards for you."</p>
"is_active"		string	Mandatory	"

Sample Request Body: (json)

```
{
  "question": "How can I create flashcards on athAlna?",
  "answer": "You can create flashcards in two ways:
Manually. Fill out the flashcard fields by entering questions and answers yourself.
Automatically. Upload your notes, and athAlna will convert them into flashcards for you.",
  "is_active": "True "
}
```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true

"message"	Indicates the actual description of the response.	String	"FAQ created successfully"
"data"	"questio": , "answer": , "is_active":,	Objects	"data": { "id": 1, "question": "How do I reset my password?", "answer": "Go to your profile and click 'Reset Password'.", "is_active": true, "created_at": "2023-10-01T12:34:56Z", "updated_at": "2023-10-02T10:15:20Z" },

Sample Response Body: (json)

c. Response Status: 200 (OK)

```
{
  "successful": true,
  "message": "FAQ created successfully.",
  "data": {
    "id": 1,
    "question": "How can I create flashcards on athAlna?",
    "answer": "You can create flashcards in two ways: Manually. Fill out the
flashcard fields by entering questions and answers yourself. Automatically.
Upload your notes, and athAlna will convert them into flashcards for you.",
    "is_active": true,
    "created_at": "2023-10-01T12:34:56Z",
    "updated_at": "2023-10-01T12:34:56Z"
  }
}
```

d. Response Status: 400 (Bad Request)

```
{
  "successful": false,
  "message": "FAQ could not be created, please try again.",
  "errors": {
```

```

    "question": ["Please provide a question"],
    "answer": ["Please provide an answer"]
  }
}

```

2. Update FAQ

- This API allows changing of existing FAQ.

Endpoint: <http://localhost:8000/faq/<int:id>/>

HTTP Method: PATCH

Request Body: (json)

Key Name	Description	Data Type	Mandatory/ Optional	Sample Value
"question"	Updating an existing FAQ Question	String	Mandatory	"How can I create flashcards on athAlna?"
"answer"	Updating an existing FAQ Answer	String	Mandatory	<p>"You can create flashcards in two ways:</p> <p>Manually. Fill out the flashcard fields by entering questions and answers yourself.</p> <p>Automatically. Upload your notes, and athAlna will convert them into flashcards for you."</p>
"is_active"	Whether the FAQ should be displayed to users	Boolean	Mandatory	"True"

Sample Request Body: (json)

```

{
  "message": "FAQ updated successfully.",
  "data": {

```

```

    "id": 1,
    "question": "How do I reset my password?",
    "answer": "Go to your profile and click 'Reset Password'.",
    "is_active": true,
    "created_at": "2023-10-01T12:34:56Z",
    "updated_at": "2023-10-02T10:15:20Z"
  },
  "successful": true
}

```

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully updated or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"FAQ updated successfully.
"data"	"questio": , "answer": , "is_active": ,	Array of Objects	{ "id": 1, "question": "How do I reset my password?", "answer": "Go to your profile and click 'Reset Password'.", "is_active": true, "created_at": "2023-10-01T12:34:56Z", "updated_at": "2023-10-02T10:15:20Z" },

Sample Response Body: (json)

e. Response Status: 200 (OK)

```
{
  "message": "FAQ updated successfully.",
  "data": {
    "id": 1,
    "question": "How do I reset my password?",
    "answer": "Go to your profile and click 'Reset Password'.",
    "is_active": true,
    "created_at": "2023-10-01T12:34:56Z",
    "updated_at": "2023-10-02T10:15:20Z"
  },
  "successful": true
}
```

f. Response Status: 400 (Bad Request)

```
{
  "successful": false,
  "message": "FAQ could not be updated, please try again."
  "error": "serializer.errors"
}
```

2. FAQ Delete

- This API allows removal of product in the database.

Endpoint: <http://localhost:8000/faq/<int:id>/>

HTTP Method: DELETE

Request Body:(json)

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"FAQ deleted successfully. "
"data"	Contains the deleted FAQ details	Objects	<pre> "data": { "id": 1, "question": "How do I reset my password?", "answer": "Go to your profile and click 'Reset Password'." , "is_active": true, "created_at": "2023-10-01T 12:34:56Z", "updated_at": "2023-10-02T 10:15:20Z" }, </pre>

Sample Response Body: (json)

a. Response Status: 200 (OK)

- Request to delete the product in the database is successfully completed.

```

"data": {
  "id": 1,
  "question": "How do I reset my password?",

```

```
"answer": "Go to your profile and click 'Reset Password'.",
"is_active": true,
"created_at": "2023-10-01T12:34:56Z",
"updated_at": "2023-10-02T10:15:20Z"
},
```

b. Response Status: 400 (Bad Request)

- Product does not exist in the database.

```
{
  "successful": false,
  "message": "FAQ could not be deleted, please try again"
}
```

3.View All FAQs

- This API allows viewing of information regarding all FAQ in the database.

Endpoint: <http://localhost:8000/products/list/>

HTTP Method: GET

Request Body: N/A

Response Body: (json)

Key Name	Description	Data Type	Sample Value
"successful"	Indicates whether the request is successfully processed or not.	boolean	true
"message"	Indicates the actual description of the response.	String	"FAQs retrieved successfully."
"data"	Array of objects containing FAQ details	Array of Objects	{ "question": "How can I create flashcards on athAlna?", "answer": "You can

			<p>create flashcards in two ways:</p> <p>Manually. Fill out the flashcard fields by entering questions and answers yourself.</p> <p>Automatically. Upload your notes, and athAlna will convert them into flashcards for you.”,</p> <p>“is_active”: " " }</p>
--	--	--	--

Sample Response Body: (json)

a. Response Status: 200 (OK)

```
{
  "successful": true,
  "message": "FAQs retrieved successfully.",
  "data": [
    {
      "id": 1,
      "question": "How can I create flashcards on athAlna?",
      "answer": "You can create flashcards in two ways: Manually. Fill out the
flashcard fields by entering questions and answers yourself. Automatically.
Upload your notes, and athAlna will convert them into flashcards for you.",
      "is_active": true,
      "created_at": "2023-10-01T12:34:56Z",
      "updated_at": "2023-10-01T12:34:56Z"
    },
  ]
}
```

b. Response Status: 404

```
{  
    "successful": false,  
    "message": "No FAQs found."  
    "data" : []  
}
```