Phishing Email Detection API Documentation

Introduction:

Welcome to the Phishing Email Detection API! This API allows users to classify emails as phishing or legitimate, retrieve classification results, and manage email records in the database. It is designed to help users identify phishing emails using AI models, enhancing email security and user awareness.

Base URL: http://127.0.0.1:5000

#1. Classify Emails:

URL: /classify
Method: POST
Description: Analyze the content of an email to determine its phishing likelihood.
Request Parameters: None

Request Body Object Structure:

```
{
  "email_subject": "string",        // Subject of the email
  "email_body": "string"            // Body content of the email
}
```

Request Body Example:

```
{
  "email_subject": "Verify Your Account",
  "email_body": "Click this link to reset your password: http://example.com"
}
```

Response Object Structure:

```
{
  "email_id": "integer",          // Unique ID of the email
  "classification": "string",     // Classification result (e.g., phishing or legitimate)
  "confidence_score": "float",    // Confidence level of the classification
  "issues": ["string"]            // List of issues detected
}
```

Response Example:

```
 {
  "email_id": 1,
  "classification": "phishing",
  "confidence_score": 0.95,
  "issues": ["Suspicious link detected"]
}
```

HTTP Status Codes:

- 201 Created: Email successfully classified.
- 400 Bad Request: Invalid or incomplete request data.
- 500 Internal Server Error: Error during classification.

---

2. Get API Status

URL: /status
Method: GET
Description: Retrieve the current status of the API and its components.
Request Parameters: None

Response Object Structure:

```
{
  "status": "string",                // Status of the API ( 'running')
  "uptime": "string",                  // Uptime of the API
  "total_emails_processed": "integer" // Number of emails processed
}
```

Response Example:

```
{
  "status": "running",
  "uptime": "24 hours",
  "total_emails_processed": 2
}
```

HTTP Status Codes:

- 200 OK: Status retrieved successfully.
- 500 Internal Server Error: Unable to fetch status.

#3. Delete Email by ID

URL: /delete/<email_id>
Method: DELETE
Description: Remove a specific email from the database using its unique ID.
Request Parameters: email_id (Unique identifier of the email)

Response Object Structure:

```
{
  "email_id": "integer",        // Unique ID of the deleted email
  "message": "string"           // Confirmation message
}
```

Response Example:

```
{
  "email_id": 1,
  "message": "Email with ID 1 has been deleted."
}
```

HTTP Status Codes:

- 200 OK: Email deleted successfully.
- 404 Not Found: Email with the specified ID does not exist.
- 500 Internal Server Error: Error during deletion.