YERIKALAGARI RAMA CHANDRA REDDY

Reg. No.:- 11905791

Contact No.:-9347825509

Question 1 Solution

1) To create an API that lists the title and description based on the category passed as an input parameter, you could use the following steps:

Use the requests library in Python to send a GET request to the endpoint https://api.publicapis.org/entries

Use the json() method to parse the response and extract the relevant data

Use the filter() function to filter the entries based on the category passed as an input parameter

Return the title and description of the filtered entries

2) To create an API that saves a new entry with all the relevant properties which retrieves values from the endpoint GET /entries, you could use the following steps:

Use the requests library in Python to send a GET request to the endpoint https://api.publicapis.org/entries

Use the json() method to parse the response and extract the relevant data

Use the user input to construct new data

Use the requests library to send a POST request to the endpoint https://api.publicapis.org/entries with the new data as the payload

Return a confirmation message indicating that the entry has been saved

3) When creating/consuming an API to ensure that it is secure and reliable, the following are some key things to consider:

Authentication and Authorization: Ensure that only authorized users can access the API and that the API can verify the identity of the user

Input validation: Ensure that all input is valid and in the expected format

Rate limiting: Limit the number of requests that can be made by a user or IP address to prevent abuse

Error handling: Ensure that the API can handle errors gracefully and return meaningful error messages

Caching: Use caching to improve performance and reduce the number of requests to the server

Monitoring: Monitor the API's performance and logs to detect any issues and troubleshoot problems.

```
<?php

class Database

{

 protected $connection = null;

 public function __construct()

 {

 try {

 $this->connection = new mysqli(DB_HOST, DB_USERNAME,
DB_PASSWORD,  DB_DATABASE_NAME);


 if ( mysqli_connect_errno()) {

 throw new Exception("Could not connect to database.");   }

 } catch (Exception $e) {

 throw new Exception($e->getMessage());

 }

 }

 public function select($query = "" , $params = [])

 {

 try {

 $stmt = $this->executeStatement( $query , $params );

 $result = $stmt->get_result()-

>fetch_all(MYSQLI_ASSOC);  $stmt->close();

 return $result;

 } catch(Exception $e) {

 throw New Exception( $e->getMessage() );

 }

 return false;

 }

 private function executeStatement($query = "" , $params = [])  {
```

```
try {
$stmt = $this->connection->prepare( $query );
if($stmt === false) {
throw New Exception("Unable to do prepared statement: " . $query);  }
if( $params ) {
$stmt->bind_param($params[0], $params[1]);
}
$stmt->execute();
return $stmt;
} catch(Exception $e) {
throw New Exception( $e->getMessage() );
}
}
}
```

Question 2 Solution

1) To tackle the challenge of reading a CSV file with data that includes both values and formulas, I would use a combination of the following steps:

Read the CSV file using a library such as pandas or csv

Iterate through each cell in the file, and for each cell that is a formula, use a library such as python expr or eval() to evaluate the formula and replace the formula with its calculated value

Write the modified data to a new CSV file

2) Some types of errors that I would check for include:

Incorrectly formatted formulas (e.g. missing operators or parentheses)

Reference to non-existent cells in formulas

Incorrectly formatted CSV file (e.g. missing or extra commas)

Issues with reading or writing the CSV file (e.g. file not found or permission denied)

3) A user could break the code by:

Providing a CSV file with a format that is not expected or supported

Providing formulas that reference cells that are not present in the file

Providing formulas that contain malicious code (e.g. code that deletes files or sends data to external servers)

Providing a CSV file with a very large number of formulas or cells, causing the program to run out of memory or take a long time to execute.