Here's a detailed explanation of each part of the PHP script, including the functions, syntax, and their roles:

**1. Request Method Check**

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

* **$\_SERVER['REQUEST\_METHOD']**: A superglobal variable that holds the HTTP request method (e.g., GET, POST, etc.).
* **===**: A strict equality operator to ensure both value and type match (POST as a string in this case).
* **Purpose**: Ensures the script runs only for HTTP POST requests, which are typically used for form submissions.

**2. Database Connection**

$conn = mysqli\_connect('localhost', 'root', '', 'web\_project');

* **mysqli\_connect()**: Establishes a connection to a MySQL database.
  + Parameters:
    - 'localhost': The server address (local machine here).
    - 'root': The MySQL username.
    - '': The MySQL password (empty here).
    - 'web\_project': The database name.
* **Purpose**: Connects the PHP script to the database for executing queries.

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

* **!$conn**: Checks if the connection failed (returns false if unsuccessful).
* **die()**: Stops script execution and outputs the provided message.
* **mysqli\_connect\_error()**: Retrieves the error message if the connection fails.
* **Purpose**: Handles database connection errors gracefully.

**3. Retrieving Input Data**

$appointment\_ids = $\_POST['appointment\_ids'];

$files = $\_FILES['files'];

* **$\_POST**: A superglobal array that holds form data sent via POST.
* **$\_FILES**: A superglobal array containing file upload data, organized as:
  + $\_FILES['files']['name']: The original file names.
  + $\_FILES['files']['tmp\_name']: Temporary paths for uploaded files.
  + $\_FILES['files']['error']: Upload error codes.
* **Purpose**: Retrieves submitted data (appointment\_ids and file details) from the form.

**4. Loop Through Appointments**

foreach ($appointment\_ids as $appointment\_id) {

* **foreach**: Iterates over arrays.
* **$appointment\_ids**: An array of appointment IDs submitted via the form.
* **$appointment\_id**: Represents the current value in the iteration.
* **Purpose**: Processes each appointment ID individually.

**5. File Upload Handling**

if (isset($files['name'][$appointment\_id]) && $files['error'][$appointment\_id] === UPLOAD\_ERR\_OK) {

* **isset()**: Checks if a variable is set and not null.
* **UPLOAD\_ERR\_OK**: A predefined constant (0) indicating a successful file upload.
* **&&**: Logical AND operator.
* **Purpose**: Verifies that:
  1. A file exists for the current appointment\_id.
  2. The file was uploaded successfully without errors.

**6. Security Measures**

**Prevent Directory Traversal**

$file\_name = basename($files['name'][$appointment\_id]);

* **basename()**: Extracts the base name of a file path.
* **Purpose**: Prevents directory traversal attacks (e.g., ../../etc/passwd) by removing directory references.

**Restrict File Types**

$allowed\_types = ['pdf', 'png', 'jpg'];

$ext = strtolower(pathinfo($file\_name, PATHINFO\_EXTENSION));

* **pathinfo()**: Extracts file path information. PATHINFO\_EXTENSION retrieves the file extension.
* **strtolower()**: Converts the string to lowercase.
* **Purpose**: Extracts the file's extension for validation against allowed types.

if (!in\_array($ext, $allowed\_types)) {

die("File type not allowed.");

}

* **in\_array()**: Checks if a value exists in an array.
* **Purpose**: Ensures only files with allowed extensions (pdf, png, jpg) are accepted.

**7. File Upload Path**

$upload\_dir = \_\_DIR\_\_ . '/../uploads/';

$file\_path = $upload\_dir . $file\_name;

* **\_\_DIR\_\_**: A magic constant representing the directory of the current script.
* **../uploads/**: A relative path pointing to a directory one level up and then to uploads/.
* **Purpose**: Defines the destination directory for uploaded files.

**8. Move Uploaded File**

if (move\_uploaded\_file($tmp\_name, $file\_path)) {

* **move\_uploaded\_file()**: Moves a file from its temporary location to a specified directory.
  + Parameters:
    - $tmp\_name: Temporary file path (from $\_FILES).
    - $file\_path: Target file path.
* **Purpose**: Saves the file securely on the server.

**9. Database Update**

$file\_url = '/myPart/uploads/' . $file\_name;

* Constructs a URL for the uploaded file, making it web-accessible.

$sql = "UPDATE appointment\_history SET file\_path = ? WHERE appointment\_id = ?";

$stmt = mysqli\_prepare($conn, $sql);

mysqli\_stmt\_bind\_param($stmt, 'si', $file\_url, $appointment\_id);

mysqli\_stmt\_execute($stmt);

mysqli\_stmt\_close($stmt);

* **mysqli\_prepare()**: Prepares a SQL statement for execution.
* **mysqli\_stmt\_bind\_param()**: Binds variables to the SQL statement placeholders (?).
  + 'si': Specifies parameter types (s for string, i for integer).
* **mysqli\_stmt\_execute()**: Executes the prepared statement.
* **mysqli\_stmt\_close()**: Frees the prepared statement resources.
* **Purpose**: Updates the database to store the uploaded file's URL for the corresponding appointment.

**10. Close Database Connection**

mysqli\_close($conn);

* **mysqli\_close()**: Closes the database connection.
* **Purpose**: Frees up server resources after database operations are complete.

**11. Redirect to Referrer**

header("Location: {$\_SERVER['HTTP\_REFERER']}");

exit;

* **header()**: Sends an HTTP header to the browser.
* **$\_SERVER['HTTP\_REFERER']**: Contains the URL of the referring page.
* **exit**: Stops further script execution.
* **Purpose**: Redirects the user back to the page they came from.

This step-by-step breakdown explains not only what each part of the code does but also how the functions and syntaxes work together to securely process file uploads and update the database.