Legacy PHP Booking System - Code Analysis

Issues Found in legacy_booking.php

1. **SQL Injection**
    - Queries directly concatenated user input (`$booking_id`, `$booking['email']`).
    - Malicious users could manipulate the database.

2. **No Input Validation**
    - `$_GET['id']` and `$_GET['site_id']` were not validated.
    - Could lead to invalid queries or errors.

3. **Poor Error Handling**
    - Database connection errors and query failures were not handled.
    - Failures would break the application silently.

4. **Code Quality Issues**
    - Repeated database connection logic.
    - Mixed logic and output in a single script.
    - Poor variable naming (`$con2`).

5. **Performance**
    - `mysqli_fetch_array` used without `MYSQLI_ASSOC`, returning unnecessary numeric keys.
    - Multiple connections opened/closed unnecessarily.

Solutions Implemented in refactored_booking.php

1. **Security**
    - Used **prepared statements** with `bind_param`.
    - Validated input parameters (`FILTER_VALIDATE_INT`).

2. **Error Handling**
    - Added `http_response_code` for invalid input or database errors.
    - Checked connection errors before executing queries.

3. **Code Quality**
    - Clear variable names: `$siteDb`.
    - Logic separated from output (still returns JSON in same format).
    - Used functions (`getSiteDbConfig`) consistently.

4. **Performance**
    - Used `fetch_assoc()` to retrieve only associative arrays.
    - Reduced unnecessary database calls and closed connections properly.

5. **Backward Compatibility**
    - Maintained same URL interface:
      ```
      refactored_booking.php?id=1&site_id=1
      ```

~~~
  ```
- JSON output format remains the same.
~~~