Python Developer Assignment

Overview:

You have been tasked with creating an API endpoint that determines the valid moves for a given chess piece on a chessboard. The chessboard contains pieces such as Rook, Queen, Bishop, and Knight, and their positions are provided as input (Refer Figure-1 for more information on positions). Your goal is to create a Dockerized application and expose an API endpoint that accepts a slug representing the chess piece and returns the valid moves for that piece. The positions of the chess pieces are given as in input in the request body in JSON format.

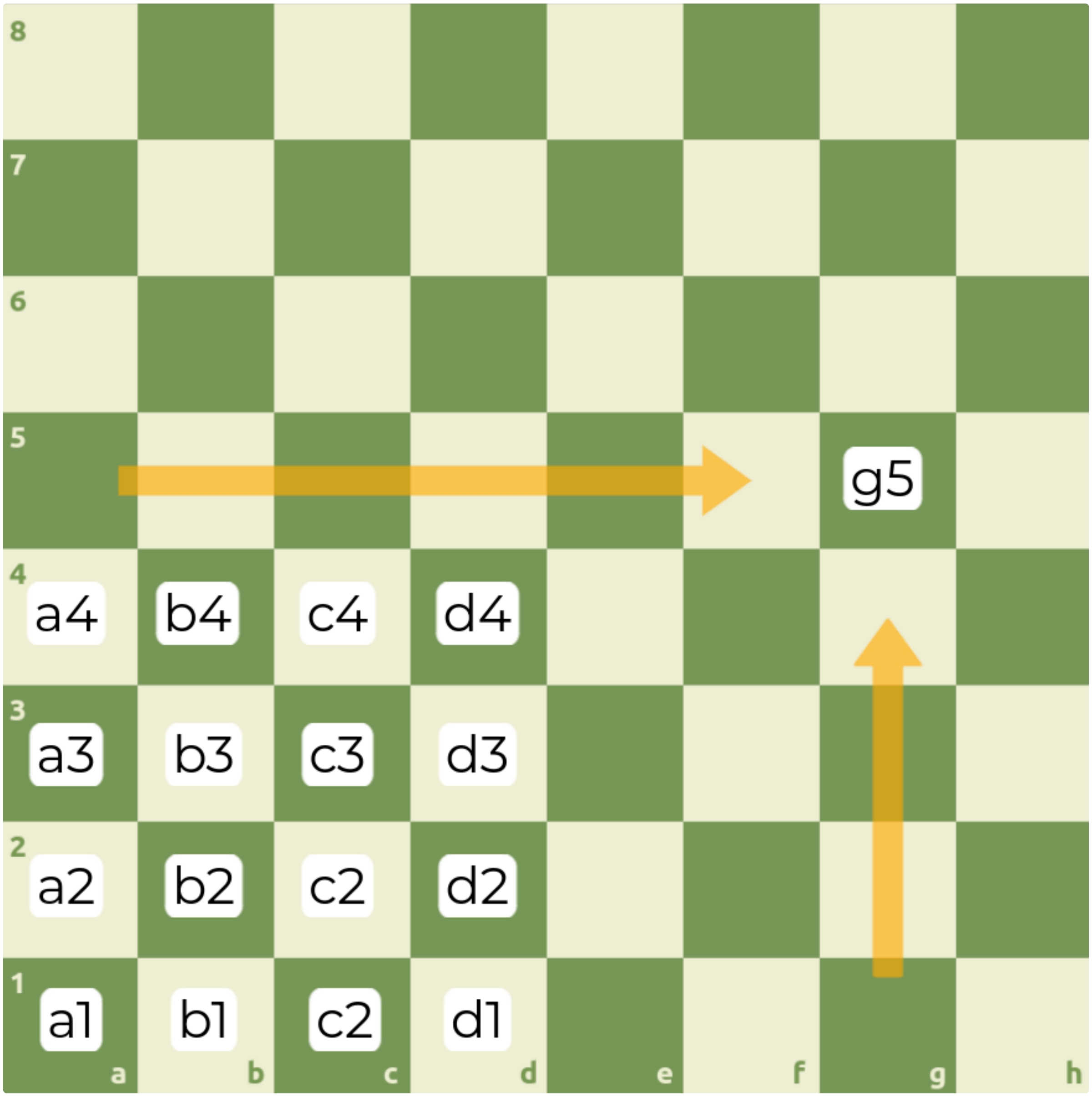

Figure-1: Chess Notations

Requirements:

1. **Dockerize the application:** Containerize the entire application using Docker to ensure easy deployment and portability.
2. **API endpoint format**: The API endpoint should follow the format: `local-host:8000/chess/<slug>`, where <slug> represents the name of the chess piece.

3. **Input and Output Examples:** Provide input and output examples to showcase the expected behavior of the API endpoint. The input should be in JSON format, and the output should be a JSON response containing the valid moves for the given chess piece.
4. **Bitbucket Repository:** Create a Bitbucket repository for the project and provide access to the email address `abhilash@infilect.com.`

Input Output Examples

Note: In the Figures the marked red spaces are invalid moves.

Example 1:

- **Input JSON:**

```
{"postions": {"Queen": "E7", "Bishop": "B7", "Rook":"G5", "Knight": "C3""}}
```

  - The Input is a dictionary with piece name as key and value is the position. In this input ‘Queen’ is in postion ‘E7’ as shown in Figure-2.

- **Endpoint:** /chess/knight
- **Output**:

```
{"valid_moves": ["A4", "A2", "B1","D1"]}
```

  - The knight can move to postions A4, A2, B1 and D1. If it moves to other places like D5 or B5 it can be captured and is not considered a valid move.

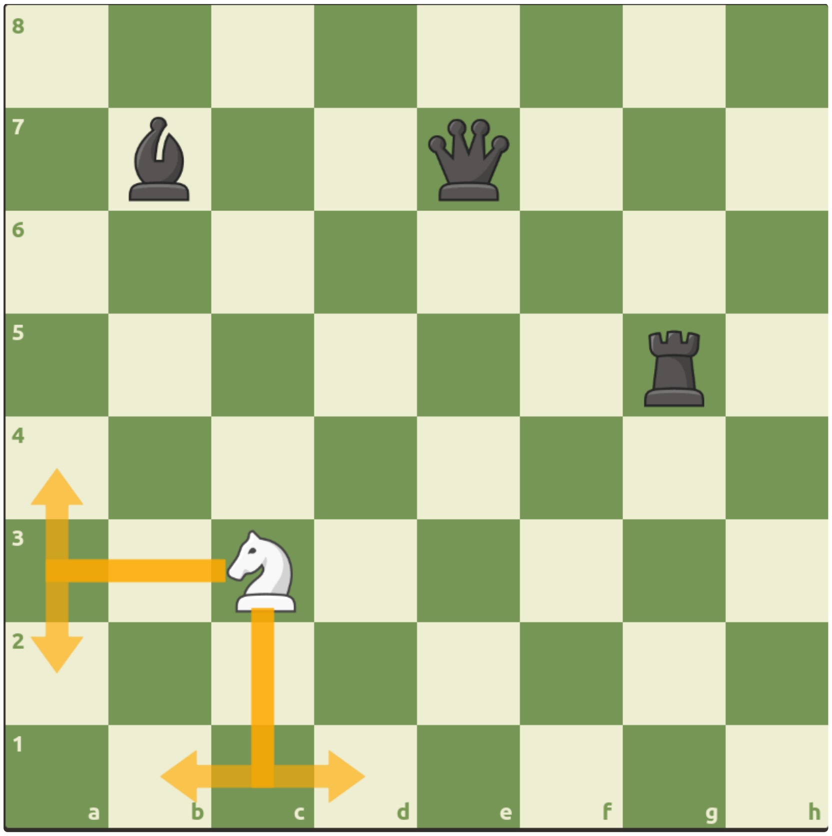

Figure-2: Ex1 Chess Positions

Example 2:

- **Input JSON:**

```
{"postions": {"Queen": "H1", "Bishop": "B7", "Rook":"H8", "Knight": "F3"}}
```

- **Endpoint**: /chess/queen
- **Output**:
  - H8 and B7 are capture moves and are considered valid.

```
{"valid_moves":["A1", "B1", "C1", "E1", "F1", "G1", "B7", "H8"]}
```

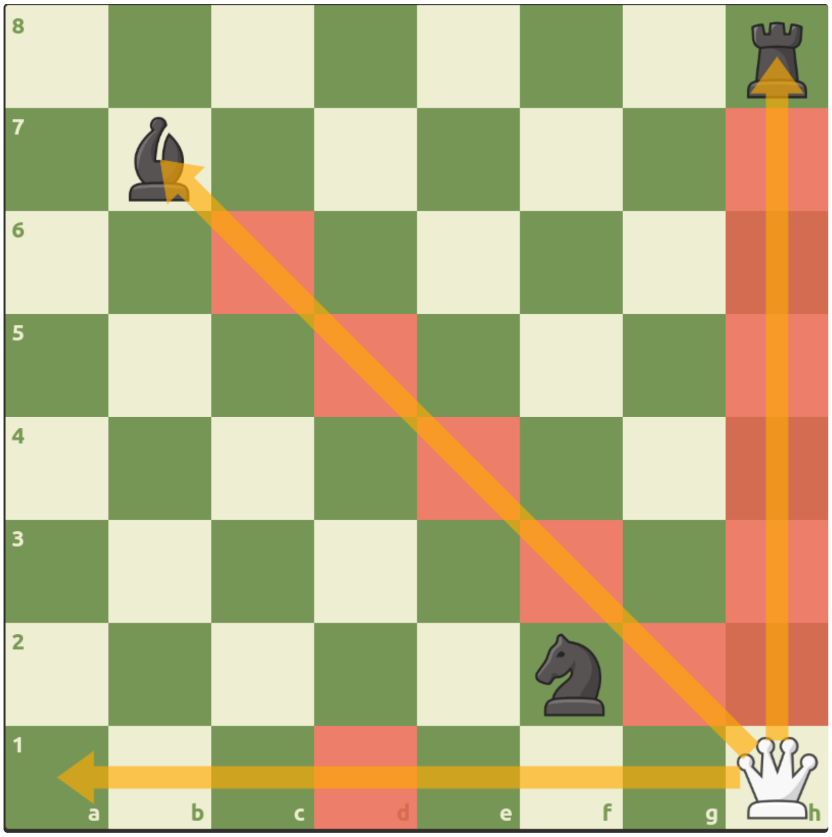

Figure-3: Ex2 Chess Positions

Example 3:

- **Input JSON:**

```
{"postions": {"Queen": "A5", "Bishop": "G8", "Rook":"H5", "Knight": "G4"}}
```

- **Endpoint**: /chess/rook
- **Output**:

```
{"valid_moves":["H1", "H3", "H4", "H8", "A8"]}
```

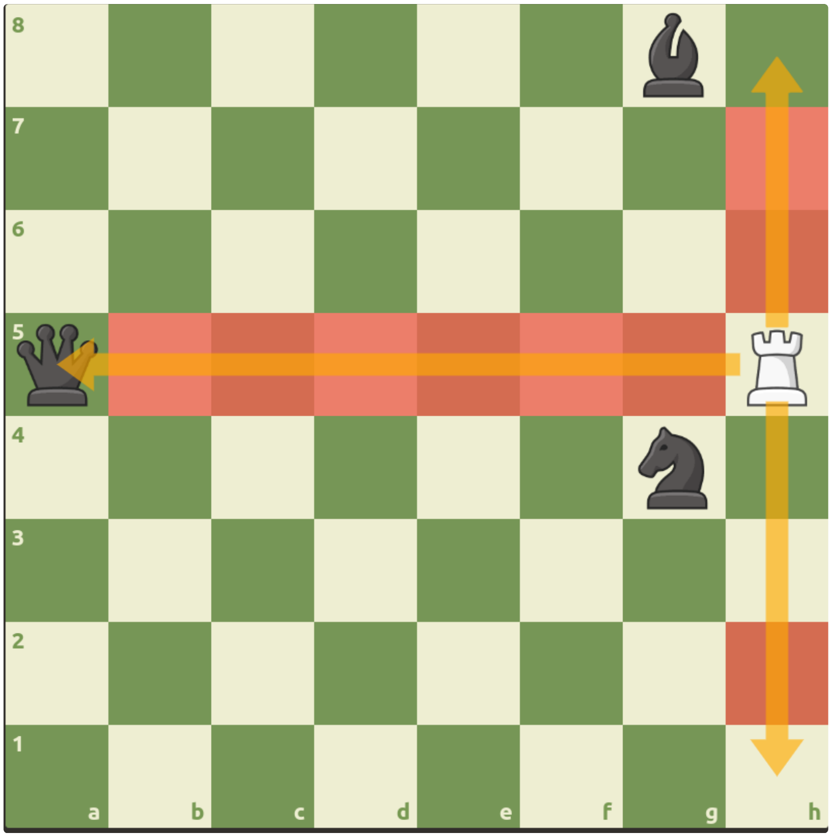

Figure-4: Ex3 Chess Positions

Submission Instructions:

1. Dockerize the application and ensure it runs successfully in a Docker container.
2. Create a Bitbucket repository for the project.
3. Share the repository access with the email address `abhilash@infilect.com`.
4. Provide the Dockerfile and any additional configuration files required for the project.
5. Include documentation or instructions on how to build, run, and test the application.

Note:

Please ensure that the API endpoint returns the valid moves for the specified chess piece based on the given positions. Take into consideration the rules and restrictions of each chess piece when calculating the valid moves.