HBnB API - Testing and Validation Report

Overview

This document summarizes the testing and validation process for the HBnB API.

It includes:

- Manual testing using curl and Swagger UI
- Input validation
- Automated tests using unittest
- Results and edge case coverage

Manual Tests via curl

Users:

Create valid user:

```
curl -X POST http://localhost:5000/api/v1/users/ -H "Content-Type: application/json" -d '{"first_name": "John", "last_name": "Doe", "email": "john@example.com"}'
```

Expected: 201 Created

Invalid user:

```
curl -X POST http://localhost:5000/api/v1/users/ -H "Content-Type: application/json" -d '{"first_name": "", "last_name": "", "email": "invalid"}'
```

Expected: 400 Bad Request

Places:

Valid place:

```
curl -X POST http://localhost:5000/api/v1/places/ ...
```

Expected: 201 Created

Invalid place:

```
curl -X POST ... with bad latitude/price
```

Expected: 400 Bad Request

Reviews:

Valid review:

```
curl -X POST http://localhost:5000/api/v1/reviews/ ...
```

Expected: 201 Created

List reviews for a place:

```
curl http://localhost:5000/api/v1/places/<PLACE_ID>/reviews
```

Expected: List response

Swagger Documentation

Accessible at: http://localhost:5000/api/v1/

Models and parameters are well documented.

Unit Testing

Command used: python3 -m unittest discover tests

Result:

```
................
----------------------------------------------------------------------
Ran 16 tests in 0.296s

OK
```

Tests included:

- User creation and validation

- Place creation and validation

- Review creation and linking

- Serialization with to_dict

Validation Summary

User:

- first_name, last_name, and email are required

- Email must be valid

Place:

- title required

- price must be positive

- latitude between -90 and 90

- longitude between -180 and 180

Review:

- text required

- user_id and place_id must reference valid records

Conclusion

All endpoints were tested, validated, and documented.

API meets all requirements:

- Input validation

- RESTful structure

- Manual + automated testing

- Swagger integration