

## Backend Assignment: API Development Task

### Assignment Title:

Develop any Two APIs Based on Provided  
Postman Collection [KPA\\_form\\_data.postman\\_collection.json](https://kpa.suvidhaen.com/KPA_form_data.postman_collection.json)  
hosted at <https://kpa.suvidhaen.com>

API documentations:  
[https://app.swaggerhub.com/apis/sarvasuvidhaen/kpa-form\\_data/1.0.0](https://app.swaggerhub.com/apis/sarvasuvidhaen/kpa-form_data/1.0.0)

login by phone number and password **7760873976 / to\_share@123**

---

### Objective:

To assess your backend development skills by implementing two APIs from a given Postman collection. The APIs should be fully functional and adhere to the specified request/response structure.

---

### Assignment Description:

You are provided with a Postman collection containing sample API requests and expected response formats for various features. Your task is to choose **any two APIs** from the collection and implement them using a backend framework of your choice (preferably **Python FastAPI**, **Django**).

---

### Key Requirements:

#### 1. API Selection:

- Review the Postman collection and choose **any 2 APIs** to implement from [https://app.swaggerhub.com/apis/sarvasuvidhaen/kpa-form\\_data/1.0.0](https://app.swaggerhub.com/apis/sarvasuvidhaen/kpa-form_data/1.0.0)
- Ensure the chosen APIs are meaningful and demonstrate different types of functionality (e.g., POST + GET, or one with file upload, another with database operations).

## 2. API Functionality:

- Implement endpoints exactly as shown in the Postman examples (URL, method, body, and response format).
- DB: Preferable to use PostgreSQL.

## 3. Documentation & Testing:

- Provide the updated Postman collection with your implemented API endpoints.
- Test your APIs using Postman to ensure they return appropriate status codes and expected responses.

## 4. Tech Stack (Preferred):

- Python (FastAPI / Django REST Framework)
- 

## Submission Requirements:

- Submit your code via GitHub or as a zip file.  
Screen record your assignment(either 2.a or 2.b. or 2.c) submission explaining your project features, technologies and stacks.
- Upload the Recording to your Drive.
- Rename your video file with your name as prefix in it. (e.g: rohan\_flutter\_assignment)  
Note: If multiple links are required, paste the links corresponding to their Label/Title
- Example:  
project-features : [https://drive.com/rohan\\_flutter\\_assignment\\_1.mp4](https://drive.com/rohan_flutter_assignment_1.mp4)  
project-technical : [https://drive.com/rohan\\_flutter\\_assignment\\_2.mp4](https://drive.com/rohan_flutter_assignment_2.mp4)
- Include a README file explaining:
  - How to set up the project.
  - Key features implemented.
  - Any limitations or assumptions made.

---

**Deliverables:**

- Source code hosted on GitHub or provided as a zipped folder.
  - Updated Postman Collection (with working API responses).
  - A short README file including:
    - Setup instructions
    - Tech stack used
    - List of implemented APIs and their descriptions
- 

**Optional (Bonus):**

- Add simple input validation (e.g., required fields, data types).
  - Use environment-based configuration (e.g., .env file for DB or base URL).
  - Swagger/OpenAPI integration (if using FastAPI).
- 

**Evaluation Criteria:**

- Functional correctness of API implementation
  - Adherence to API request/response structure
  - Code clarity and modularity
  - Postman demonstration with working responses
  - Completeness of documentation
- 

**Submission Requirements:**

**Mail the details at [contact@suvidhaen.com](mailto:contact@suvidhaen.com)**

- Submit code via GitHub or Google Drive zip link.
- Include:

- source-code: [https://drive.com/yourname\\_api\\_assignment.zip](https://drive.com/yourname_api_assignment.zip)
  - postman-collection: [https://drive.com/yourname\\_postman\\_collection.json](https://drive.com/yourname_postman_collection.json)
  - readme: [https://drive.com/yourname\\_readme.txt](https://drive.com/yourname_readme.txt)
- 

**Deadline: 3 Days from the date of assignment receipt**