Sure! Here's an example of an overview documentation for a codebase, using a stub generator application with a Flask backend and a React frontend as a reference:

---

# Code Base Overview Documentation

## Project Title: Stub Generator Application

### Introduction

The Stub Generator Application is designed to intercept API calls from an application, store responses under subcomponent names, and retrieve stored data when dependencies are down. The application includes a form to input data and integrates seamlessly into existing systems for API calls.

### Table of Contents

1. [Project Structure](#project-structure)

2. [Technologies Used](#technologies-used)

3. [Setup and Installation](#setup-and-installation)

4. [Backend: Flask](#backend-flask)

5. [Frontend: React](#frontend-react)

6. [Interception Mechanism](#interception-mechanism)

7. [Deployment](#deployment)

8. [Contributing](#contributing)

9. [License](#license)

### Project Structure

```

stub-generator-application/

│

├── backend/

│ ├── app.py

│ ├── requirements.txt

│ ├── routes/

│ │ ├── \_\_init\_\_.py

│ │ ├── api\_routes.py

│ │ └── stub\_routes.py

│ ├── models/

│ │ ├── \_\_init\_\_.py

│ │ └── stub\_model.py

│ └── utils/

│ ├── \_\_init\_\_.py

│ └── interceptor.py

│

├── frontend/

│ ├── public/

│ │ ├── index.html

│ ├── src/

│ │ ├── components/

│ │ │ ├── App.js

│ │ │ ├── Form.js

│ │ │ └── StubList.js

│ │ ├── services/

│ │ │ └── apiService.js

│ │ ├── App.js

│ │ ├── index.js

│ │ └── styles/

│ │ └── app.css

│ ├── package.json

│ └── webpack.config.js

│

└── README.md

```

### Technologies Used

- \*\*Backend\*\*: Flask, Python

- \*\*Frontend\*\*: React, JavaScript

- \*\*Database\*\*: SQLite (for simplicity, can be upgraded to any RDBMS)

- \*\*Deployment\*\*: Azure App Service, Docker

### Setup and Installation

#### Prerequisites

- Python 3.x

- Node.js and npm

- Docker (optional for containerized deployment)

#### Backend Setup

1. Navigate to the `backend/` directory:

```bash

cd backend

```

2. Create a virtual environment:

```bash

python -m venv venv

```

3. Activate the virtual environment:

- On Windows:

```bash

venv\Scripts\activate

```

- On macOS/Linux:

```bash

source venv/bin/activate

```

4. Install the dependencies:

```bash

pip install -r requirements.txt

```

5. Run the Flask application:

```bash

flask run

```

#### Frontend Setup

1. Navigate to the `frontend/` directory:

```bash

cd frontend

```

2. Install the dependencies:

```bash

npm install

```

3. Start the React application:

```bash

npm start

```

### Backend: Flask

The Flask backend handles API requests, stores data, and serves the stubbed responses. Key components include:

- \*\*app.py\*\*: The main entry point for the backend application.

- \*\*routes/\*\*: Contains route definitions for API and stub-related endpoints.

- \*\*models/\*\*: Defines data models.

- \*\*utils/\*\*: Utility functions, including the API call interceptor.

### Frontend: React

The React frontend provides a user interface for managing stubs. Key components include:

- \*\*components/\*\*: Contains reusable UI components like forms and lists.

- \*\*services/\*\*: API service to communicate with the Flask backend.

### Interception Mechanism

The interceptor module captures API calls, logs them, and either forwards them to the intended endpoint or serves a stubbed response. This is crucial for testing and development when dependencies are unavailable.

### Deployment

Deploy the application to Azure App Service or any other cloud platform. For containerized deployment, a `Dockerfile` is provided.

#### Docker Deployment

1. Build the Docker image:

```bash

docker build -t stub-generator-app .

```

2. Run the Docker container:

```bash

docker run -p 5000:5000 stub-generator-app

```

### Contributing

1. Fork the repository.

2. Create a new branch (`git checkout -b feature/your-feature-name`).

3. Commit your changes (`git commit -am 'Add some feature'`).

4. Push to the branch (`git push origin feature/your-feature-name`).

5. Create a new Pull Request.

### License

This project is licensed under the MIT License.

---

Feel free to customize this template according to your specific project details and requirements.