# BCI Online Processing Backend ReadMe

This is a Django-based server project named mysite. This document will guide you through the project structure and its features.

## Project Structure:

mysite:

This is the main project folder containing all global routes and configuration files.

## Submodules:

### bciDB:

* Manages the database on the server.
* Modify the database structure by editing custom classes in models.py.

### chat:

* An application for maintaining Websocket long connections.
* Main functionality processing logic is in consumers.py, which can be modified to adjust how the server receives, saves, and processes online data.

### data\_analysis:

* An application for offline data analysis.
* Includes an independent file upload page; modify offline data processing methods in views.py.

### dataset:

* Used for data generation and caching, and saving paths when receiving online data streams.

### static:

* Directory for storing JavaScript static files.

### temp:

* Directory for storing temporary files.

## Installation and Running:

### Docker Deployment:

### 1 Build Docker Image:

* In the root directory of the project, create a file named Dockerfile and include the following content:

Bash：

FROM python:3.8

ENV PYTHONUNBUFFERED 1

RUN mkdir /code

WORKDIR /code

COPY requirements.txt /code/

RUN pip install -r requirements.txt

COPY . /code/

* This sets up the base image, installs necessary Python dependencies, and copies project files from the current directory to the /code directory in the image.

### 2 Build and Run Container:

* Build the Docker image using the following command:

docker build -t mysite .

* Run the Docker container:

docker run -d -p 8000:8000 mysite