PassFort

Wiki Coding Task

We'd like you to create the backend for a wiki, like Wikipedia. With the following requirements:

1. A wiki is a collection of **documents**
2. **Documents** are lumps of plain text. No graphics, attachments, or formatting
3. Each **document** is uniquely identified by a **title** that is a maximum of 50 characters in length. This **title** does not change for the life of a document
4. A **document** can have multiple **revisions**, as it is updated over time. We store all historical **revisions** of a document
5. We should be able to view the **document** as *it was* at any point in time. I.e. we can use any timestamp to fetch a **revision** e.g. If we have a document at time 1pm and time 3pm, then sending a timestamp of 2pm should return the document as it was at time 1pm.

Your task is to implement a JSON api with the following endpoints:

`GET /documents`
This should return a list of available titles.

`GET /documents/<title>`
This should return a list of available revisions for a document.

`GET /documents/<title>/<timestamp>`
This should return the document as it was at that timestamp.

`GET /documents/<title>/latest`
This should return the current latest version of the document.

`POST /documents/<title>`
This allows users to post a new revision of a document.
It should receive JSON in the form: `{content: 'new content...'}`.

Technical implementation requirements:

- We ask that you spend **only 2.5 hours** on this task. Please start the task by initialising a git repository locally. At the end of this timeframe create a git bundle by running `git bundle create passfort.bundle master` and send this to us (this is to ensure all candidates are evaluated fairly).
- The code should be production ready; it should have error handling
- You should write some automated tests around your application
  - It is up to you to decide which tests and how to write them

You can use any technologies you like. Our current stack is using `python` with `flask`, but you are welcome to choose anything you like. Feel free to use a starter kit; but please put a link to it in a README file in your task.

How you'll be assessed:

We're not expecting complete implementation, and the degree of completion will vary around experience level. We're more interested in you showcasing best practices and attention to detail, rather than completing the whole task poorly. Treat it like you would any production code.