Password Keeper/Manager App

Statement for Candidates:

Below is the problem statement for your project.

1. Your task is to create an **API** that supports below project requirements.
2. You have 3 hours to develop the API.
3. You are free to use google to find solutions, however copying an existing project is not allowed.
4. At the end of the 3 hour time period, you will be given get 15 minutes to do your submission on Google Form & to push your project in a GitHub repository, submission form: https://forms.gle/g8AePuVet1LWnjLv8
5. Your API submission will be evaluated by our team and a result will be published at 4:00 PM on the same day.
6. After submission, you can do an *optional* activity of building front-end (UI) for this project. By doing this activity you will be evaluated for your UI skills along with your back-end API submission. The front-end task is mentioned at the end of the problem statement.
7. You will have 1.5 hour to develop the mentioned UI requirements. Feel free to user any frontend technology (HTML, CSS, JS / Android / IOS / Flutter / react / vue). submission form: https://forms.gle/zEsr5jSyDAfz6BC19

What is the project?
A web application that keeps record of users' passwords. User can log into the panel to see username and passwords of websites they have saved.

Requirements:

1. User account registration:
   Create a user account. These credentials will be used to log into this panel.

```
[POST] /app/user

Request Data: {
    'username': str,
    'password': str
}

Response Data: {
    'status': 'account created'
}
```

2. User account login:
   Provide ability to log into the panel using the user credentials.

```
[POST] /app/user/auth

Request Data: {
    'username': str,
    'password': str
}

Response Data: {
    'status': 'success',
    'userId': int
}
```

3. List of saved usernames and passwords for different websites:
   Provide ability for user to see list of previously stored website usernames & passwords.

   ```
   [GET] /app/sites/list/?user={userId}

   Request Data: None
   Response Data: [List of stored website username & passwords]
   ```

   List returned should be belong to the `userId` passed with the request

4. Save a new username & password for website:
   Provide ability for user to add new username & password for a website.

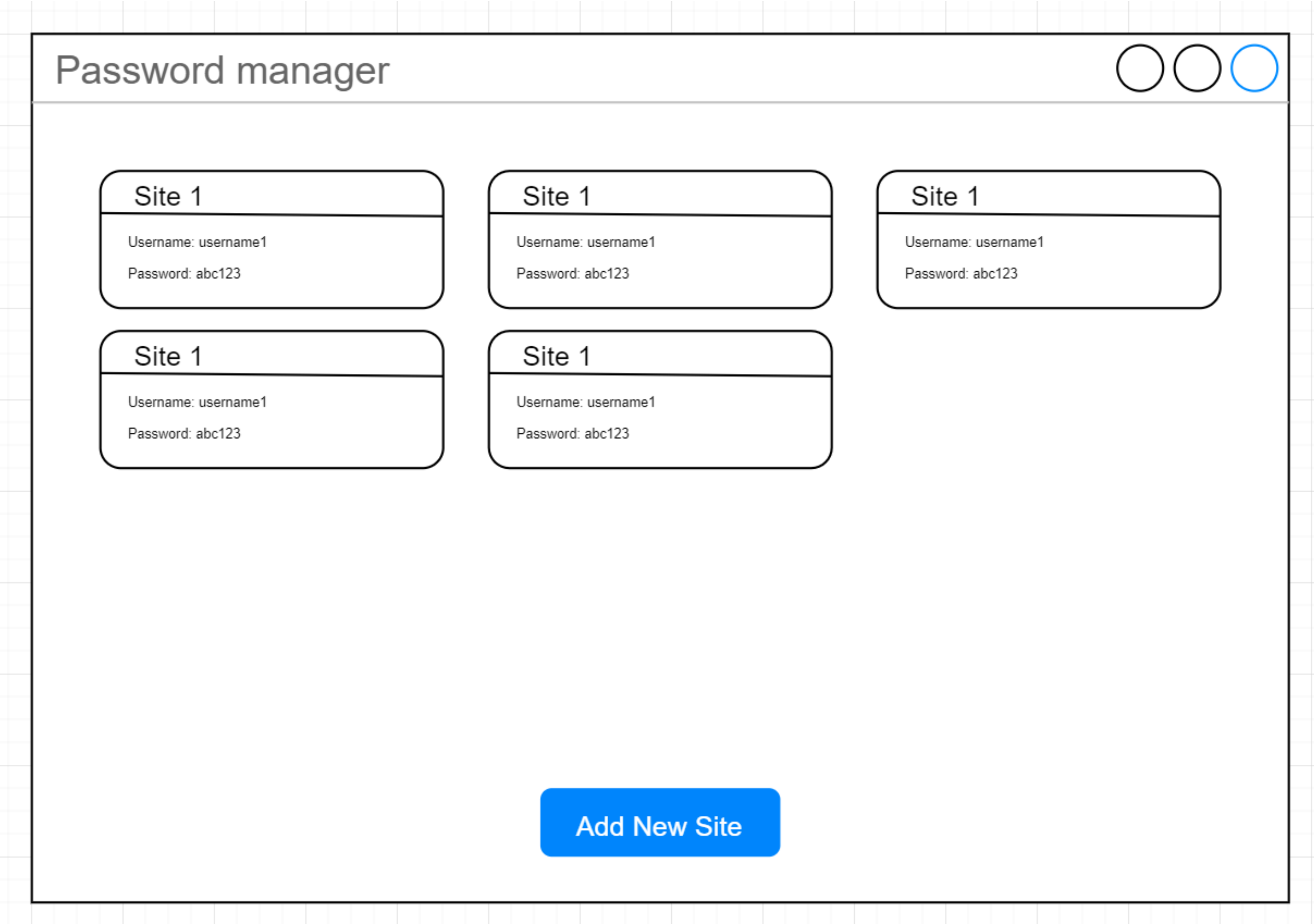
   ```
   [POST] /app/sites?user={userId}

   Request Data: {
       'website': str,
       'username': str,
       'password': str
   }

   Response Data: {
       'status': 'success'
   }
   ```

5. **Mandatory Requirement:** All the passwords stored in the database have to be encrypted.
   No one should be able to read the stored passwords without a secret key. The key can be stored in the code itself.

Optional Front-end Requirements:

Build a static page that showcases your UI/UX skills.
(Note: No need to connect your UI with backend, invest your 1.5 hour only for UI/UX)

1. Build UI for showing list of stored website usernames and passwords. [*As mentioned in requirement point 3*]
2. Build UI for the form which will take input for adding new website username and passwords. [*As mentioned in requirement point 4*]



Password manager

Site 1

Username: username1

Password: abc123

Site 1

Username: username1

Password: abc123

Site 1

Username: username1

Password: abc123

Site 1

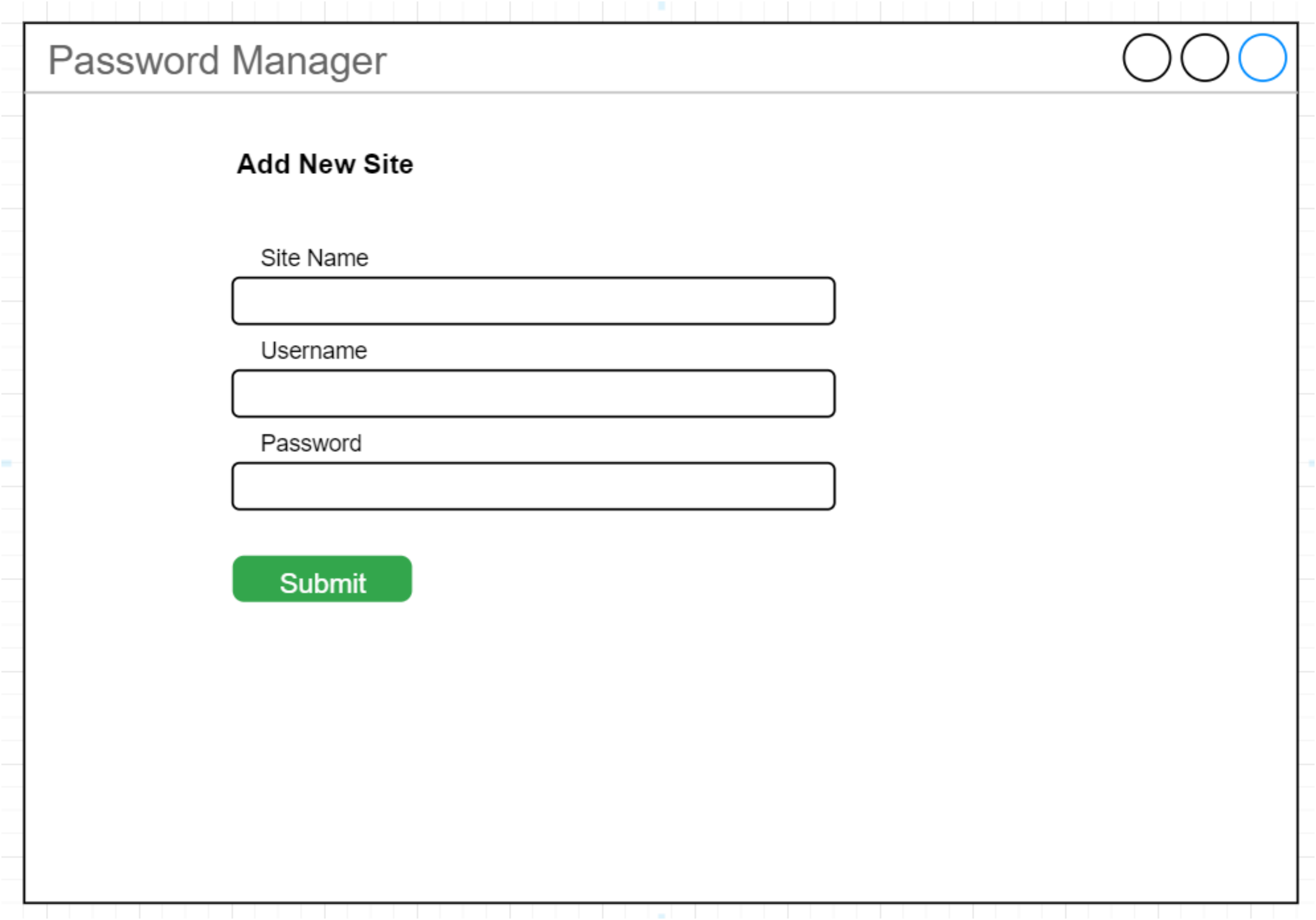
Username: username1

Password: abc123

Site 1

Username: username1

Password: abc123



Tech Stack:

1. Any web-server of your choice (Python Flask / Django, NodeJS Express / Koa, Java, etc)
2. Database: MySQL (Compulsory)