Practice Problem Workshop 9:
Part I - Individual Practice Questions

INFO 2310, Spring 2026

Student ID: Solution **NetID:**

Overview

Retrieval practice is the act of actively recalling information from memory, like quizzing oneself.

This workshop is designed to help you prepare and study for upcoming exams by engaging in retrieval practice.

Part I: Individual Practice Questions

Credit: 10 points. (completion)

1. Open this booklet and try your best to answer the practice problems **individually**.

   If you need to refer to your notes, you can. But you should try to answer the questions on your own first, without looking at your notes. (This will help you prepare for the upcoming exam.)

2. Once a majority of the class has answered the practice questions on their own, **form groups of 3-4 people and go over the questions together**.

   Discuss the questions and share your answers with the group.

   You should update your answers in this booklet if necessary.

   Discussing the answers will help you practice retrieving the information from memory and will also help you learn from your peers.

Submit this booklet for completion credit.

Practice Problem 1

Pretend that you're a web browser and that the user wants to view the HTML file below. Write **each** HTTP **request** that your browser would make to the server to retrieve all the resources needed to display the webpage.

`index.html`:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/styles/site.css">
  </head>
  <body>
    <h1>Cornell University Gallery</h1>
    <section>
      <ul>
        <li><img src="/images/mcgraw-tower.jpg" alt="McGraw Tower"></li>
        <li><img src="/images/uris-library.jpg" alt="Uris Library"></li>
        <li><img src="/images/bailey-hall.jpg" alt="Bailey Hall"></li>
      </ul>
    </section>
    <script src="/scripts/modal.js"></script>
  </body>
</html>
```

Your HTTP requests should be in the proper **order** and correct HTTP packet **format**.

GET /index.html HTTP/1.1

GET /styles/site.css HTTP/1.1

GET /images/mcgraw-tower.jpg HTTP/1.1

GET /images/uris-library.jpg HTTP/1.1

GET /images/bailey-hall.jpg HTTP/1.1

GET /scripts/modal.js HTTP/1.1

Practice Problem 2

Implement a REST API endpoint in Express.js to look up a user's profile information based on their **username** from the following MongoDB `users` collection.

```
[
  {
    "username": "ezra",
    "name": "Ezra Cornell",
    "email": "ezra.cornell@cornell.edu"
  },
  ...
]
```

Employ the design principles of RESTful APIs and return appropriate HTTP status codes for successful and unsuccessful requests. Only implement the endpoint (you may omit the boilerplate code for setting up the Express.js server).

```
app.get('/api/users/:login', async (req, res) => {
    const login = req.params.login
    const doc = await app.locals.db.collection('users')
        .findOne({ username: login })
    if (doc) {
        res.status(200).json(doc)
    } else {
        res.status(404).send()
    }
})
```