# CS 499 Module One Assignment Template

**Tyler Dickerson**

1. **Self-Introduction:** Address all of the following questions to introduce yourself.
   1. How long have you been in the Computer Science program?

**I have been in the Computer Science program for a little over two years.**

* 1. What have you learned while in the program? List three of the most important concepts or skills you have learned.

**Understanding the stages of software development from planning and design to implementation and maintenance.**

**Problem solving when things aren’t going the way I wanted or planned them to.**

**Database Management: Learning to design, implement, and manage databases using SQL and NoSQL technologies.**

* 1. Discuss the specific skills you aim to demonstrate through your enhancements to reach each of the course outcomes.

**I aim to demonstrate my ability to design and develop efficient software solutions and manage complex databases. Specifically, I will showcase skills in data structure optimization, and advanced database management.**

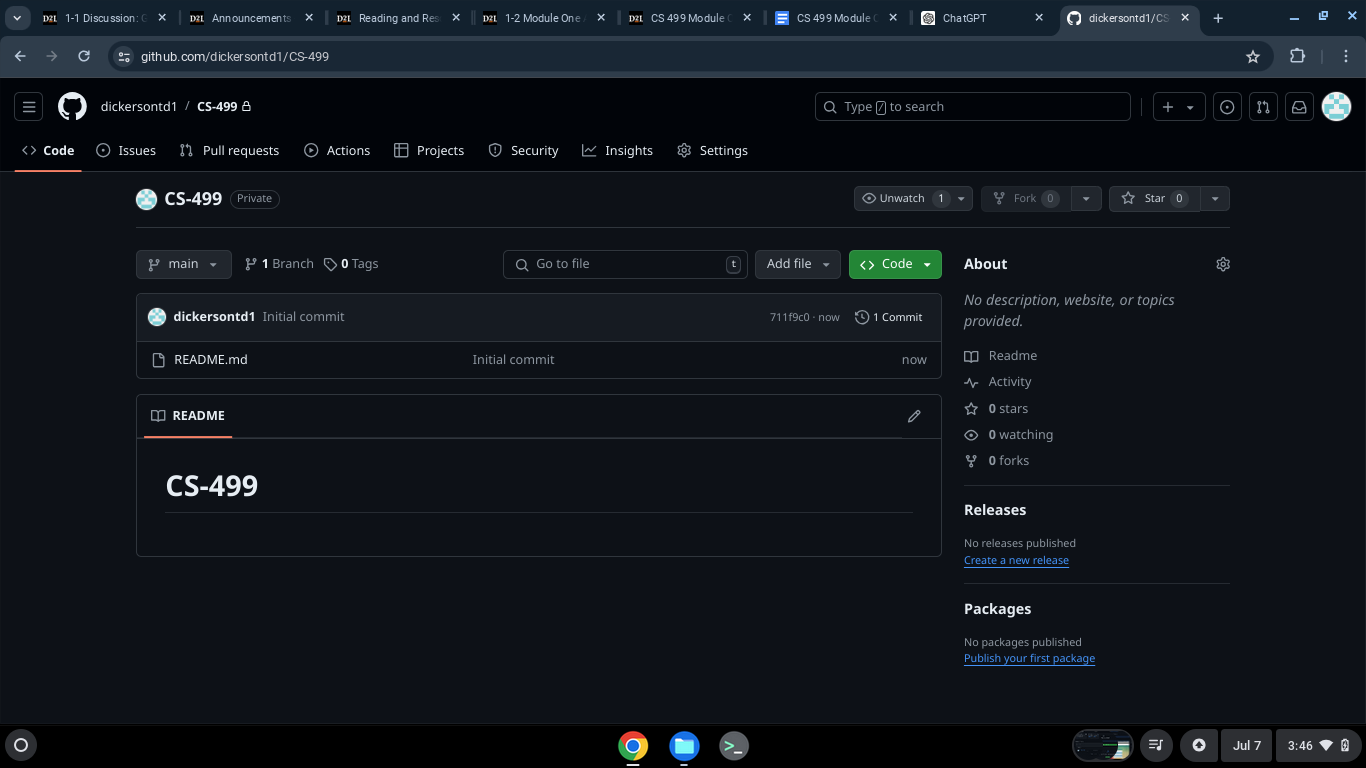
* 1. How do the specific skills you will demonstrate align with your career plans related to your degree?

**These skills align with my career plans to become a worker in the computer science field, though I do not know what I exactly want to be, I know I will most likely need to design, develop, and maintain both front-end and back-end systems efficiently.**

* 1. How does this contribute to the specialization you are targeting for your career?

**These enhancements will help my future career by demonstrating my proficiency in software engineering, algorithm optimization, and database management, which are crucial for developing applications.**

1. **ePortfolio Set Up:**
   1. Submit a **screen capture** of your ePortfolio GitHub Pages home page that clearly shows your URL.
      1. You already have a repository in GitHub where you uploaded projects in previous courses. Your ePortfolio will reside in GitHub but can link to work at other sites, such as Bitbucket.
   2. Use the GitHub Pages link in the Resource section for directions on:
      1. How to create your GitHub website and publish code to GitHub Pages
      2. Issues, such as adding links to other sites
   3. Paste a screenshot of your GitHub Pages home page with your URL clearly showing in the space below.

****

### **Enhancement Plan**

#### **Category One: Software Engineering and Design**

**Selected Artifact:  
Artifact: Project utilizing MongoDB for data processing and storage.  
Origin: Developed as part of CS 340: Advanced Programming Concepts.**

**Enhancement Plan:**

**Objective:  
Enhance the project by incorporating advanced features, optimizing data processing pipelines, ensuring scalability, and implementing security best practices.**

**Planned Enhancements:**

1. **Advanced Query Optimization:**
   * **Create indexes on critical fields to speed up query performance.**
   * **Implement query optimization techniques using MongoDB’s advanced features.**
2. **Enhanced Data Security:**
   * **Use encryption for data storage and transmission to protect sensitive information.**
   * **Implement role-based access control (RBAC) for better security management.**
   * **Implement JWT-based authentication to secure API endpoints and ensure only authorized users can access the system.**
3. **Scalability and Performance:**
   * **Deploy on Kubernetes for better scalability and management of resources.**
   * **Use MongoDB’s sharding and replica set features to enhance performance and ensure high availability.**

**Pseudocode/Flowchart:**

**# Connect to MongoDB**

**Connect to MongoDB with username and password**

**# Create database and collection**

**Create a new database called 'enhanced\_db'**

**Create a new collection called 'data\_collection'**

**# Create indexes for faster queries**

**Create index on 'field1' in ascending order**

**Create index on 'field2' in descending order**

**# Insert data with encryption**

**Insert data into the collection with encryption**

**# Run advanced query**

**Find documents where 'field1' is 'value1'**

**Group the results by 'field2' and count them**

**# Deploy on Kubernetes**

**Deploy the application and MongoDB on Kubernetes**

**Database Setup:**

* **Connect to MongoDB.**
* **Create/encrypt collection.**

**Indexing and Query Optimization:**

* **Create indexes.**
* **Define optimized query pipeline.**

**Deployment:**

* **Write Kubernetes deployment scripts.**
* **Deploy MongoDB instance on Kubernetes.**

**Skills and Course Outcomes:**

**Skills to Demonstrate:**

* **Advanced database management with MongoDB.**
* **Implementing security best practices including JWT-based authentication.**
* **Deploying on Kubernetes for scalability.**

**Course Outcomes Alignment:**

* **Design and evaluate computing solutions: Designing scalable and secure architecture using MongoDB and Kubernetes.**
* **Demonstrate innovative techniques and tools: Using MongoDB’s advanced features and Kubernetes for deployment.**
* **Develop a security mindset: Ensuring data security and privacy with encryption, RBAC, and JWT authentication.**

#### **Category Two: Algorithms and Data Structures**

**Selected Artifact:  
Artifact: Project implementing data processing algorithms using MongoDB. animalshelter1.py  
Origin: Developed as part of CS 340: Advanced Programming Concepts.**

**Enhancement Plan:**

**Objective:  
Enhance the project by implementing advanced data structures and optimizing algorithms for high-throughput data processing.**

**Planned Enhancements:**

1. **Advanced Data Structures:**
   * **Implement a Distributed B-Tree for efficient range queries across large datasets.**
   * **Use Bloom filters for fast existence checks to reduce unnecessary database queries.**
2. **Algorithm Optimization:**
   * **Optimize data ingestion algorithms for higher throughput.**
   * **Implement parallel processing for data aggregation to handle large volumes of data more efficiently.**

**Pseudocode/Flowchart:**

**plaintext**

**Copy code**

**# Define a B-Tree node**

**Create a node with a list of keys and a list of children**

**# Define a B-Tree**

**Create a B-Tree with a root node**

**# Insert a key into the B-Tree**

**If the root node is full, split it and create a new root**

**Insert the key into the appropriate node**

**# Search for a key in the B-Tree**

**Start at the root and search through the keys**

**If the key is not found, move to the appropriate child node**

**Repeat until the key is found or the leaf is reached**

**# Parallel processing for data aggregation**

**Distribute data processing tasks across multiple CPU cores**

**Aggregate results from parallel tasks into a final result**

**B-Tree Setup:**

* **Initialize root node.**
* **Define insert and search methods.**

**Bloom Filter Setup:**

* **Initialize filter with size.**
* **Implement add and check methods.**

**Algorithm Optimization:**

* **Optimize data ingestion.**
* **Implement parallel processing logic.**

**Skills and Course Outcomes:**

**Skills to Demonstrate:**

* **Implementation of advanced data structures like Bloom Filters and Distributed B-Trees.**
* **Optimization of data processing algorithms using parallel processing.**
* **Efficient data management through advanced querying and data retrieval techniques.**

**Course Outcomes Alignment:**

* **Design and evaluate computing solutions: Enhancing data structures and algorithms for better efficiency.**
* **Demonstrate innovative techniques and tools: Utilizing distributed B-Trees, Bloom filters, and parallel processing to optimize data management.**

#### **Category Three: Databases**

**Selected Artifact:  
Artifact: Project with a Redis backend for session management and caching.  
Origin: Developed as part of CS 340: Advanced Programming Concepts.**

**Enhancement Plan:**

**Objective:  
Enhance the project by integrating Apache Cassandra for large-scale data storage, using Redis for caching and session management, and implementing a RESTful API with secure authentication.**

**Planned Enhancements:**

1. **Database Integration:**
   * **Connect Redis for caching and session management to improve performance.**
   * **Integrate Apache Cassandra for large-scale data storage, ensuring scalability and reliability.**
2. **Advanced Features:**
   * **Implement data replication and failover in Cassandra for high availability.**
   * **Develop a robust RESTful API for database interactions using secure JWT-based authentication.**
3. **Full-Stack Development:**
   * **Build a Node.js backend with a React frontend.**
   * **Implement JWT-based authentication for secure user management and protection of API endpoints.**

**Pseudocode/Flowchart:**

**# Connect to Redis**

**Connect to Redis on localhost**

**# Connect to Cassandra**

**Connect to Cassandra on localhost**

**# Redis caching**

**Save data to Redis with a key and value**

**Get data from Redis using a key**

**# Cassandra data storage**

**Insert data into a Cassandra table**

**Fetch data from a Cassandra table using an ID**

**# Example usage**

**Save 'value1' in Redis with key 'key1'**

**Print the value stored in Redis for 'key1'**

**Insert data with ID 1 and value 'data1' into a Cassandra table**

**Print the data fetched from Cassandra with ID 1**

**# RESTful API with JWT authentication**

**Define API endpoints for data interactions**

**Implement JWT-based authentication to secure API access**

**Database Setup:**

* **Connect to Redis and Cassandra.**

**Caching Logic:**

* **Implement caching functions.**

**Data Storage Logic:**

* **Implement data insertion and fetching functions.**

**API Development:**

* **Design RESTful API endpoints for database interactions.**

**Full-Stack Integration:**

* **Develop backend with Node.js.**
* **Create frontend with React.**

**Skills and Course Outcomes:**

**Skills to Demonstrate:**

* **Integration of multiple NoSQL databases (Redis and Cassandra).**
* **Advanced data management and caching techniques using Redis and Cassandra.**
* **Full-stack development with Node.js and React.**
* **Implementation of secure authentication using JWT.**

**Course Outcomes Alignment:**

* **Demonstrate innovative techniques and tools: Using Redis and Cassandra for efficient data management and developing a secure full-stack application.**
* **Design and evaluate computing solutions: Creating a comprehensive solution using modern technologies like Node.js, React, and JWT.**

### **ePortfolio Overall Skill Set**

**Skills and Outcomes:**

**Code Review Skills:**

* **Advanced database management with MongoDB, Redis, and Cassandra.**
* **Implementation of security best practices, including JWT-based authentication.**
* **Optimization of data processing algorithms using parallel processing and advanced data structures.**

**Narrative Skills:**

* **Designing and evaluating scalable computing solutions.**
* **Demonstrating innovative techniques and tools in database management and full-stack development.**
* **Developing a security mindset to protect data and applications.**

**Professional Self-Assessment:**

* **Reflecting on the enhancement process and the skills developed through this project.**
* **Discussing the impact of the enhancements on the overall project and how they align with industry standards.**
* **Evaluating personal growth and setting future learning goals to further develop expertise in software engineering and database management.**

**Enhancement plan-**

**Category One: Software Engineering and Design**

**Selected Artifact:**

**Artifact: Project utilizing MongoDB for data processing and storage.**

**Origin: Developed as part of CS 340: Advanced Programming Concepts.**

**Enhancement Plan:**

**Objective:**

**Enhance the project by incorporating advanced features, optimizing data processing pipelines, and ensuring scalability and security.**

**Planned Enhancements:**

**Advanced Query Optimization:**

**Create indexes on critical fields.**

**Implement query optimization techniques.**

**Enhanced Data Security:**

**Use encryption for data storage and transmission.**

**Implement role-based access control (RBAC).**

**Scalability and Performance:**

**Deploy on Kubernetes for better scalability.**

**Use MongoDB’s sharding and replica set features.**

**Pseudocode/Flowchart:**

**# Connect to MongoDB**

**Connect to MongoDB with username and password**

**# Create database and collection**

**Create a new database called 'enhanced\_db'**

**Create a new collection called 'data\_collection'**

**# Create indexes for faster queries**

**Create index on 'field1' in ascending order**

**Create index on 'field2' in descending order**

**# Insert data with encryption**

**Insert data into the collection with encryption**

**# Run advanced query**

**Find documents where 'field1' is 'value1'**

**Group the results by 'field2' and count them**

**# Deploy on Kubernetes**

**Deploy the application and MongoDB on Kubernetes**

**Database Setup:**

**Connect to MongoDB.**

**Create/encrypt collection.**

**Indexing and Query Optimization:**

**Create indexes.**

**Define optimized query pipeline.**

**Deployment:**

**Write Kubernetes deployment scripts.**

**Deploy MongoDB instance on Kubernetes.**

**Skills and Course Outcomes:**

**Skills to Demonstrate:**

**Advanced database management with MongoDB.**

**Implementing security best practices.**

**Deploying on Kubernetes for scalability.**

**Course Outcomes Alignment:**

**Design and evaluate computing solutions: Designing scalable and secure architecture.**

**Demonstrate innovative techniques and tools: Using MongoDB’s advanced features and Kubernetes.**

**Develop a security mindset: Ensuring data security and privacy.**

**Category Two: Algorithms and Data Structures**

**Selected Artifact:**

**Artifact: Project implementing data processing algorithms using MongoDB.**

**Origin: Developed as part of CS 340: Advanced Programming Concepts.**

**Enhancement Plan:**

**Objective:**

**Enhance the project by implementing advanced data structures and optimizing algorithms for high-throughput data processing.**

**Planned Enhancements:**

**Advanced Data Structures:**

**Implement a distributed B-tree for efficient range queries.**

**Use bloom filters for fast existence checks.**

**Algorithm Optimization:**

**Optimize data ingestion algorithms for higher throughput.**

**Implement parallel processing for data aggregation.**

**Pseudocode/Flowchart:**

**Pseudocode:**

**# Define a B-Tree node**

**Create a node with a list of keys and a list of children**

**# Define a B-Tree**

**Create a B-Tree with a root node**

**# Insert a key into the B-Tree**

**If the root node is full, split it and create a new root**

**Insert the key into the appropriate node**

**# Search for a key in the B-Tree**

**Start at the root and search through the keys**

**If the key is not found, move to the appropriate child node**

**Repeat until the key is found or the leaf is reached**

**B-Tree Setup:**

**Initialize root node.**

**Define insert and search methods.**

**Bloom Filter Setup:**

**Initialize filter with size.**

**Implement add and check methods.**

**Algorithm Optimization:**

**Optimize data ingestion.**

**Implement parallel processing logic.**

**Skills and Course Outcomes:**

**Skills to Demonstrate:**

**Implementation of advanced data structures.**

**Optimization of data processing algorithms.**

**Efficient data management.**

**Course Outcomes Alignment:**

**Design and evaluate computing solutions: Enhancing data structures and algorithms for better efficiency.**

**Demonstrate innovative techniques and tools: Utilizing distributed B-tree and bloom filters.**

**Category Three: Databases**

**Selected Artifact:**

**Artifact: Project with a Redis backend for session management and caching.**

**Origin: Developed as part of CS 340: Advanced Programming Concepts.**

**Enhancement Plan:**

**Objective:**

**Enhance the project by integrating Apache Cassandra for large-scale data storage, using Redis for caching and session management, and implementing a RESTful API.**

**Planned Enhancements:**

**Database Integration:**

**Connect Redis for caching and session management.**

**Integrate Apache Cassandra for large-scale data storage.**

**Advanced Features:**

**Implement data replication and failover in Cassandra.**

**Develop a robust API for database interactions using RESTful principles.**

**Full-Stack Development:**

**Build a Node.js backend with a React frontend.**

**Implement JWT-based authentication for secure user management.**

**Pseudocode/Flowchart:**

**Pseudocode:**

**# Connect to Redis**

**Connect to Redis on localhost**

**# Connect to Cassandra**

**Connect to Cassandra on localhost**

**# Redis caching**

**Save data to Redis with a key and value**

**Get data from Redis using a key**

**# Cassandra data storage**

**Insert data into a Cassandra table**

**Fetch data from a Cassandra table using an ID**

**# Example usage**

**Save 'value1' in Redis with key 'key1'**

**Print the value stored in Redis for 'key1'**

**Insert data with ID 1 and value 'data1' into a Cassandra table**

**Print the data fetched from Cassandra with ID 1**

**Database Setup:**

**Connect to Redis and Cassandra.**

**Caching Logic:**

**Implement caching functions.**

**Data Storage Logic:**

**Implement data insertion and fetching functions.**

**API Development:**

**Design RESTful API endpoints for database interactions.**

**Full-Stack Integration:**

**Develop backend with Node.js.**

**Create frontend with React.**

**Skills and Course Outcomes:**

**Skills to Demonstrate:**

**Integration of multiple NoSQL databases.**

**Advanced data management and caching techniques.**

**Full-stack development with Node.js and React.**

**Course Outcomes Alignment:**

**Demonstrate innovative techniques and tools: Using Redis and Cassandra for efficient data management.**

**Design and evaluate computing solutions: Creating a comprehensive solution using modern technologies.**

**ePortfolio Overall Skill Set**

**Skills and Outcomes:**

**Code Review Skills:**

**Advanced database management with MongoDB.**

**Implementation of security best practices.**

**Optimization of data processing algorithms.**

**Narrative Skills:**

**Designing and evaluating scalable computing solutions.**

**Demonstrating innovative techniques and tools.**

**Developing a security mindset.**

**Professional Self-Assessment:**

**Reflecting on the enhancement process and the skills developed.**

**Discussing the impact of the enhancements on the overall project.**

**Evaluating personal growth and future learning goals.**