Nayeem Bhuiyan
nyembn@gmail.com
https://github.com/nyembn
https://www.linkedin.com/in/nyembn

Education

City University of New York, Queens College
Bachelor of Science in Computer Science, June 2020

Projects

A Functional Programming Language (Java; Feb 2020 - Current; Source):
An implementation of a functional sub-language of LISP.

- Implemented a lexical analyzer that analyzes tokens of the language.
- Designed and constructed parser from grammar specification.
- Implemented expression and function evaluation from language specification.

BYKR (Python, Flask, Cloud SQL, HTML, CSS, JavaScript; Dec 2018 - Current; Source; Demo): A cycling specific web application for blogging. It lets users write details about their bicycle ride experiences. Users can set up a profile and add bicycle/s they have. Chart feature shows their ride statistics.

- Python data structures list and tuples are used to implement data structures. General algorithms for data insert, update and delete is used.
- Designed the database schema. Implemented SQL transaction to insert data on different tables simultaneously.
- Implemented statistics visualization by JSON using Chartjs.
- Currently working to implement a NoSQL persistent data solution.

OpenEMR Email Feature (Java, Model Driven Health Tools, MySQL; Feb 2016 - May 2016; Source; Team of four project): A Java extension intended to provide email functionality for OpenEMR (open-source electronic medical record and practice management application). It can be used to email patient records to other entities involved in patient care. This extension retrieves patient information from a MySQL database using JDBC. After that it converts the data in a Continuity of Care Document format using Model Driven Health Tools.

- Analyzed object oriented data model of Model Driven Health Tools.
- Implemented Model Driven Health Tools interfaces to structure patient data in Continuity of Care Document format.

Thread Synchronization (Java; Oct 2015 - Dec 2015; Source): A demonstration of synchronization of a multi threaded scenario.

- Implemented data structures using Java Collections Framework.
- Threads are implemented using Java Runnable interface and Thread class. Implemented guarded blocks idiom and monitor for synchronization.
- Used Timer and TimerTask classes to implement timing of threads.

Skills

Java, C++, Python, HTML, CSS, JavaScript, Data Structures, Algorithms, git, Google Cloud Platform