HARSHIT JAIN

+1 (582) 203-9755 ◇ harshitj.cs@gmail.com ◇ linkedin.com/in/harshitjain17 ◇ github.com/harshitjain17

EDUCATION

The Pennsylvania State University, University Park PA Aug 2021 - Dec 2024 (Expected)
Bachelor of Science in Computer Science | Minor in Mathematics
GPA: 3.75/4.0 | Dean's List (All Semesters) | Recipient of The President Walker Award | Webmaster @ NDL | Resident Assistant
Relevant Coursework: Data Structures & Algorithms, Systems Programming, Artificial Intelligence, Programming Language Concepts, Programming and Computation II: Data Structures (in Python), OOP with Web-based Applications (in Java), CodePath: Intermediate Software Engineering, Linear Programming, Discrete Mathematics

SKILLS

Programming Languages: Python, JavaScript, C/C++, Java, HTML/CSS, MATLAB, Verilog, Assembly (64/32-bit x86)
Softwares: Visual Studio, Microsoft SQL Server, MySQL Database System, Linux/UNIX, Postman, Agile, JIRA
Frameworks & Tools: GoLang, Node.js, React.js, RESTful APIs, LaTeX, Git

WORK EXPERIENCE

Software Engineering Intern, Research Associate May 2022 - Present
Materials Research Institute (MRI), The Pennsylvania State University *University Park, PA*

- Contributed to the development of DMR-FIRST Instrument Tool (NSF-funded project), a full-fledged search engine for researchers across the U.S. to locate signature research instruments at their nearest location
- Implemented front-end architecture using React.js to design 50+ latest user-facing features with 100% accuracy (tested using JEST) and built reusable components and front-end libraries for continuous development
- Tested and optimized 20+ REACT components for the best performance across every device and browser
- Integrated MS SQL Server relational database which currently deals with 500+ instruments and 18+ tables
- Saved weeks of development efforts by integrating leverageable modular code by using popular REACT libraries such as Material UI, improving the efficiency by approximately 40%
- Developed and tested Python scripts automating the process to retrieve data from various sources, manipulate and analyze the data, filter out irrelevant data, look up similar data in the server, import it into the server, and handle errors gracefully
- Collaborated and networked effectively with team members at each step of the design process to accomplish the sprint goals
- Utilized: Python, JavaScript, React.js, Node.js, MS SQL Server, RESTful APIs, Git, HTML/CSS, JIRA, Office 365

PROJECT EXPERIENCE

`mdadm` Linear Device Feb 2023 - Present

- Developed `mdadm` tool in *C* for managing multiple disks in Linux systems
- Configured 16 disks of size 64 KB as a 1 MB linear device, providing users with a unified address space for data access
- Implemented mount/unmount operations to the linear device, preventing potential data loss and system crashes
- Designed the read/write functions to set up in the linear device, providing users with comprehensive data access capabilities
- Rolled out robust error-checking mechanisms including checks for out-of-bounds access and large read-and-write operations

Course Scheduler (Try it here) Oct 2022 - Dec 2022

- Developed the application using Java and SQL to allow students and educators to manage their course schedules, including setting up semesters, adding courses, and new students
- Designed the Derby Database to store and organize the data, and wrote several SQL queries to facilitate tasks such as adding and dropping courses, and managing the waitlist process

Library Management System (Try it here) Nov 2021 - Dec 2021

- Built a Python-based professional library management system that streamlines basic and advanced tasks in a library setting; the system passed tests on 500+ library logs
- Developed multiple features and functions for the system, including:
  - Checks the eligibility of students to borrow books on a particular day for a certain number of days
  - Finds the most borrowed/popular books in the library
  - Finds the book that has the highest borrow ratio
  - Produces a sorted list of the most borrowed books (books with the highest usage ratio)
  - Calculates the pending fines at the end of the log and on a specific day in the log