

# Varshith Pabbisetty

Address: The Alameda, San Jose, CA, 95126

Mobile: +1 (214) 885-9784

Email: [krishnavarshith.pabbisetty@sjsu.edu](mailto:krishnavarshith.pabbisetty@sjsu.edu)

GitHub: <https://github.com/VarshithPabb1setty>

LinkedIn: <https://www.linkedin.com/in/krishnavarshithpabbisetty/>

## EDUCATION

---

**San Jose State University (SJSU), San Jose, CA, U.S.A** | Master of Science in Software Engineering **Aug 2023 - May 2025**  
Relevant Coursework: Software Systems Engineering, Distributed Systems, Data Mining, Machine Learning, Design and Analysis of Algorithms.

**Vellore Institute of Technology (VIT) Vellore, Tamil Nadu, India** | **Jul 2017 - Jun 2021**  
Bachelor of Technology in Computer Science & Engineering  
Relevant Coursework: Data Structures and Algorithms, Object Oriented Programming, Operating Systems, Networks, High Performance Computing, Artificial Intelligence.

## SKILLS

---

**Languages & Frameworks:** JavaScript, Java, Python, C, C++, React.js, Node.js, Express.js, Spring Boot, HTML, CSS, Boost, Material-UI  
**Tools & Paradigms:** Git, Github, VSCode, APM, REST APIs, Microservices, CUDA, OpenMP, Multi-threading, Agile Methodologies, Mocha, Chai.js, Kubernetes, Chrono  
**Databases and Technologies:** MySQL, PostgreSQL, MongoDB, AWS, Firebase, Pickle

## PROFESSIONAL EXPERIENCE

---

Software Engineer **Jio Platforms Limited** **June 2021 - July 2023** | Navi Mumbai, India

- Led backend development within an **Agile** team of 55 for the 'Re-imagined Procurement Platform', implementing **RESTful APIs** in **Node.js** and **Express.js**. Secured an **A+** performance rating.
- Enhanced application performance by 20% using **Redis** for caching and **Kafka** for real time data processing.
- Crafted comprehensive unit tests using **Mocha** and **Chai.js**, achieving over 90% code coverage and participated in code reviews to significantly enhance code quality.
- Managed **CI/CD** with **Kubernetes** for orchestration, **APM** for monitoring, and **Azure** for Task Board and deployments.
- Optimized **MongoDB** and **MySQL** databases, implementing **aggregations** and **indexing** to reduce record fetch times to under 100ms, handling thousands of records efficiently.

Software Engineering Intern **Ken42 EdTech Pvt. Ltd.** **Mar 2021 - Jun 2021** | Bengaluru, India

- Engineered 5 intuitive frontend **UI** components for a digital classroom platform, streamlining enrollments, course material access, attendance tracking, and student grading processes.
- Used **React.js** and **Material UI** to build the component library with over 140 components and integrated **Storybook** for documentation.
- Revamped web functionalities and **CMS** for course materials, integrated **Salesforce** for robust student data handling, and debugged React.js code, enhancing management efficiency for over 1,500 student profiles.

## PROJECTS

---

### Parallelization of Minmax Algorithm for Tic-Tac-Toe

- Developed an optimized Minmax algorithm for tic-tac-toe, achieving a 40% reduction in computation time by parallelizing the game tree search using **OpenMP**.
- Implemented in **C++**, enhancing decision-making efficiency and supporting advanced game strategies for **AI** opponents.
- Tech:** C++, OpenMP, Boost, Chrono, C

### Movie Theater Club App: <https://github.com/gopinathsjsu/team-project-wizard>

- Implemented **JWT Auth** using **MongoDB** as the database for 3 user types, enhancing security for members, non-members and staff.
- Created an admin dashboard for theater, show time and seat management.
- Deployed on **AWS** with **Auto Scaling** across 4 **EC2** instances and **NGINX** for load balancing.
- Tech:** Typescript, Node.js, React.js, HTML, CSS, MongoDB, AWS, Mantine

### Custom MVCC Implementation: <https://github.com/VarshithPabb1setty/CMPE-273---Implementation-of-MVCC-using-Pickle>

- Enabled over 100 concurrent data operations with a **MVCC** system, utilizing **Python** and **FastAPI**.
- Built a data structure to guarantee data isolation and consistency and **ZMQs** messaging queue for communication.
- Used **Python's Pickle** module as a data store.
- Tech:** Python, ZMQ, Pickle, Fast API, Hashlib