KALPIT **BORKAR**

kalpitborkar@gmail.com
Github | Linkedin | Portfolio

Education

Indian Institute of Technology Bombay
Bachelor of Technology in Electrical Engineering
Pursuing Minor in Computer Science

Skills

Languages: C++, Python, JavaScript, Go, Assembly, VHDL
Technologies: Node.js, Express.js, MongoDB, HTML5, CSS3, Pug, Git
ML Frameworks: TensorFlow, Keras, OpenCV, Matplotlib, Pandas

Selected Projects

Spanning Tree Protocol | [link]
• Devised an algorithm to implement spanning tree protocol in **C++** to prevent looping in the network
• Built a LAN-bridge topology to simulate a physical network with up to 26 LAN ports using OOPs
• Created a packet-tracer for debugging, tracing, and analysing packet transfers in real-time

Library Management REST API | [link]
• Implemented CRUD functionalities on server-side to access and modify data using **Node** and **Express**
• Connected server to **MongoDB** database to store various entities created using **Mongoose**
• Created 33 routes and controllers to request and update information

Python Git Implementation | [link]
• Developed Git version control system reimplementing all fundamental features of git from scratch.
• Implemented commands like add, cat-file, checkout, commit, hash-object, init, log, ls-tree, merge, rebase, rev-parse, rm, show-ref and tag

Chip-8 emulator | [link]
• Emulated Chip-8 by creating an 8-bit microcomputer virtual machine to run chip-8 programs
• Enforced opcodes, cycles, stack, audio & visuals using **C++** to build the infrastructure of chip-8
• Applied **SDL2 library** to provide low level access to keyboard to interact with the programs

Python Interpreter | [link]
• Built a virtual stack machine to manipulate several stacks and perform its operations purely in **Python**
• Implemented virtual machines, frames, functions and blocks to create the interpreter

2D Physics Engine | [link]
• Created a 2D Physics Engine to simulate particle collisions, gas cloud, soft body and springs in **C++**
• Designed algorithms to simulate particle collision, attraction, combination, acceleration, bounce, etc

Face Anti-Spoofing System | [link]
• Built a face anti-spoofing system by implementing Resnet 50 V2 feature vector using **TensorFlow**.
• Increased the accuracy of the system by 17% using data augmentation techniques and integrating Haar Cascade model in the system achieving 94.8% validation accuracy.