ResVision - Where BFT meets Clarity

Divyanshu Malik, Karamjeet Gulati, Akshit Parmar, Aditya Sharoff, Devashree Kataria

UC DAVIS
COMPUTER SCIENCE

Problem Statement

- Visualize PBFT consensus protocol, showing various communication phases on real data.
- Create intuitive and scalable visualization framework to visualize SMR protocols
- Traditional methods of teaching and comprehending PBFT rely heavily on theoretical descriptions and static illustrations.
- There exists a gap in our ability to provide an intuitive and real-time understanding of how the PBFT protocol operates, communicates, and adapts to changing network conditions.

UCDAVIS
COMPUTER SCIENCE

Benefits

- Educational Value: Enhance the learning experience by offering a visually intuitive tool for students and practitioners to comprehend the PBFT consensus protocol.

- Insightful Analysis: Enable users to observe and analyze the protocol's behavior in real-time.

- Teaching Aid: Facilitate instructors in conveying complex concepts effectively, fostering a better understanding of distributed systems and consensus algorithms.

UCDAVIS
COMPUTER SCIENCE

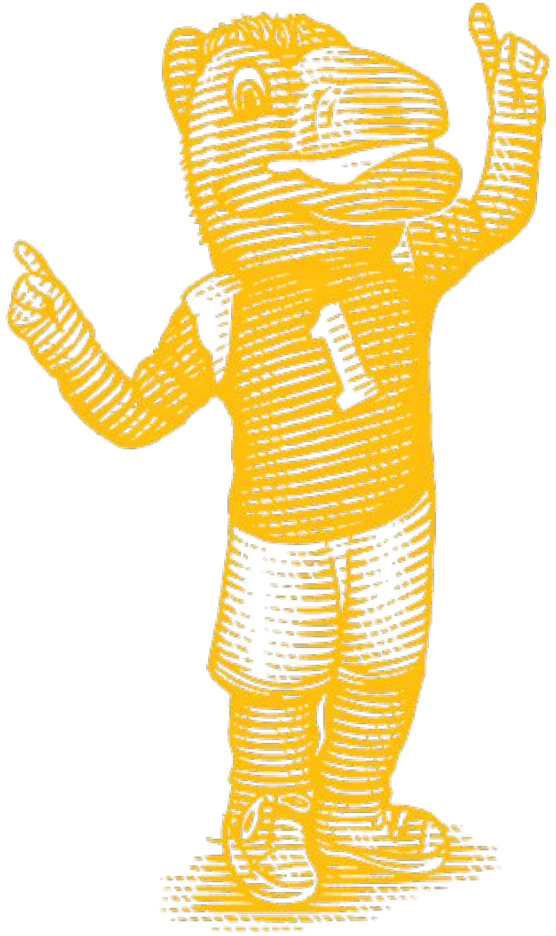
Existing approaches

- Most current visualization tools show a phase divided flow diagram
- Only PBFT is visualised
- Difficult to extend for visualising other protocols
- Cannot breakdown communication phases for better visualization

UCDAVIS
COMPUTER SCIENCE

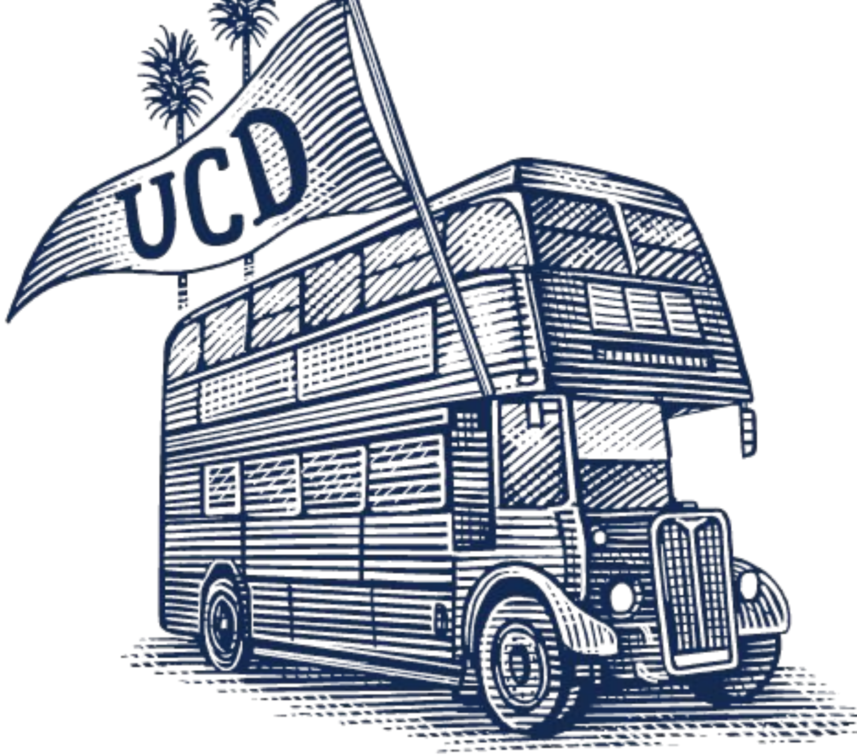
Our Approach...

- Visual representation of node communication, highlighting message flow and types.
- Dynamic display of state transitions, providing insights into the protocol's progression.
- Uses actual communication logs from ResilientDB for visualization
- Easily extendable to make a more richer visualization
- Can be used to visualize other protocols like HotStuff, Zyzzyva etc



UCDAVIS
COMPUTER SCIENCE

Technologies Used



Front-end : React.js, Css, Javascript, D3.js, Figma
Back-end : Node.js, Resilientdb, Google Cloud Platform

UCDAVIS
COMPUTER SCIENCE

Front-End

- We used react.js which is a front-end Javascript library for creating user interfaces.

- Figma designs serve as the initial blueprints for our website, providing a visual representation of the intended user interface and user experience.

- The website holds information about BFT, PBFT and the phases of PBFT

UCDAVIS
COMPUTER SCIENCE

Front End

- In order to create the BFT visualization we used D3.js for producing Dynamic interactive data visualizations in web browser. D3 makes use of Scalar Vector graphics, HTML5 and Cascading Style Sheets

- Developed a client-side WebSocket for receiving and parsing real-time data.

- We connected our D3.js visualizations with the data coming from the backend to simulate different transactions and phases included in the consensus protocol such as Pre-prepare, prepare, commit, and response.

UC DAVIS
COMPUTER SCIENCE

Backend

- Developed Bash scripts for automating key processes:
  - Downloading Resilient DB.
  - Running Docker containers.
  - Copying files between containers and local directories.
  - Stopping and starting containers.

- Streamlined interactions with Resilient DB for increased efficiency.

UCDAVIS
COMPUTER SCIENCE

Back-End

- Developed a parser for server log files, specifically focusing on transaction data.

- Implemented a server-side WebSocket server integrated with Resilient DB.

- Established a file system monitoring component for observing log file changes.

Pretty PBFT Attempts

- Pretty PBFT uses dummy data rather actual data
- Changed File path to implement our custom parsed data
- We ran into issues getting it to accept custom data even when we formatted it in such a manner that was acceptable to the front end
- Ultimately, we were ABLE to fully realize the Pretty PBFT made by our classmates last year with real time data

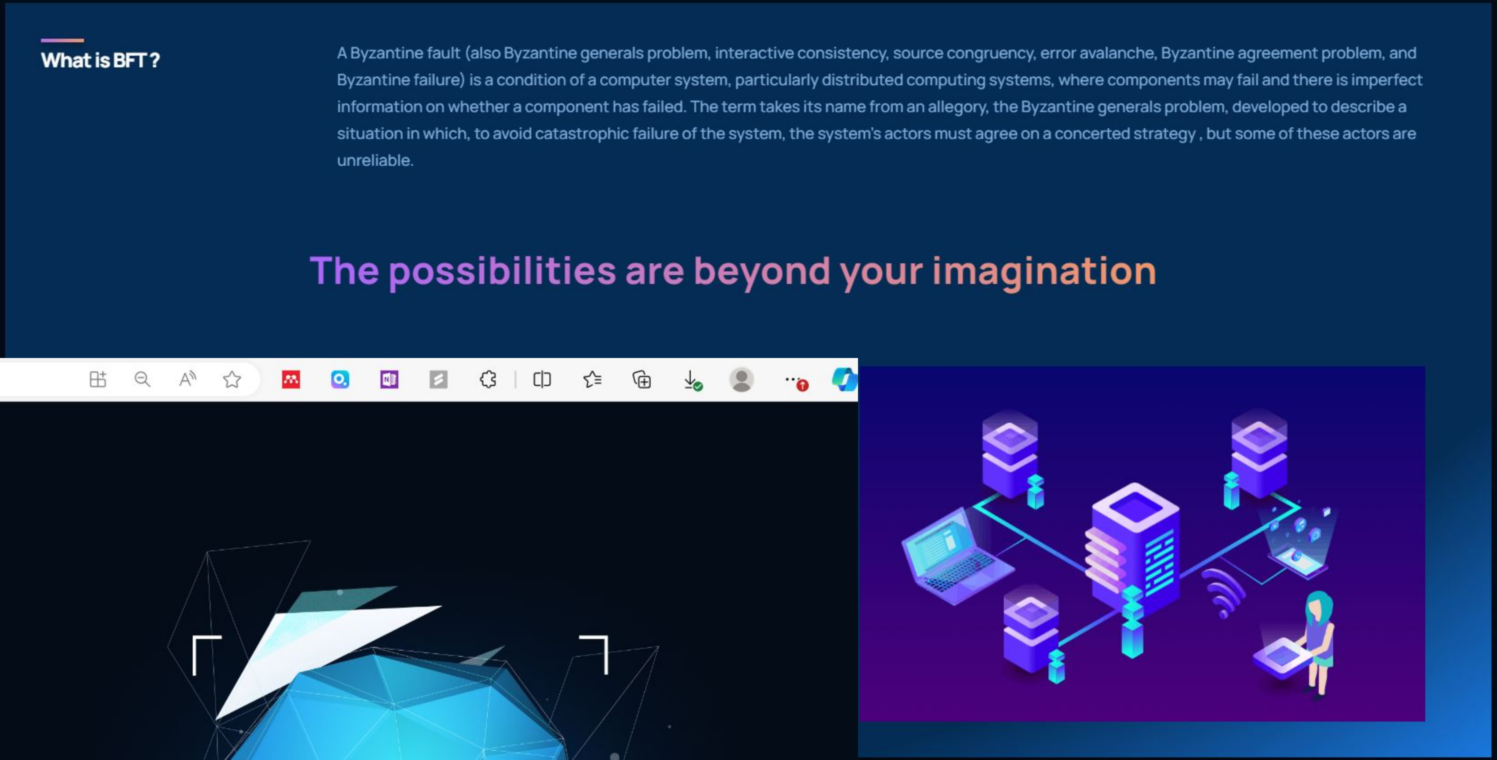
UCDAVIS
COMPUTER SCIENCE

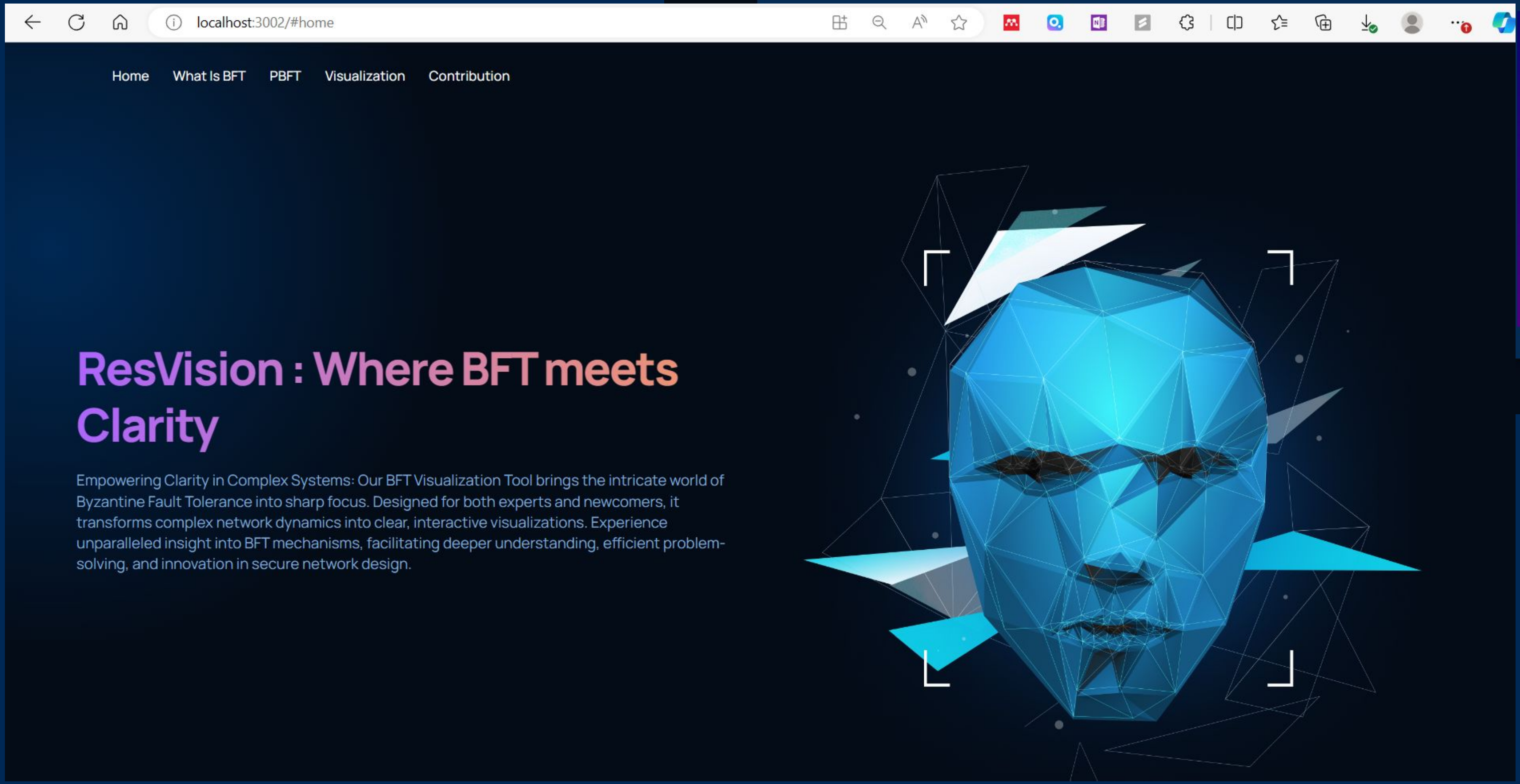
Demo

React App
React App

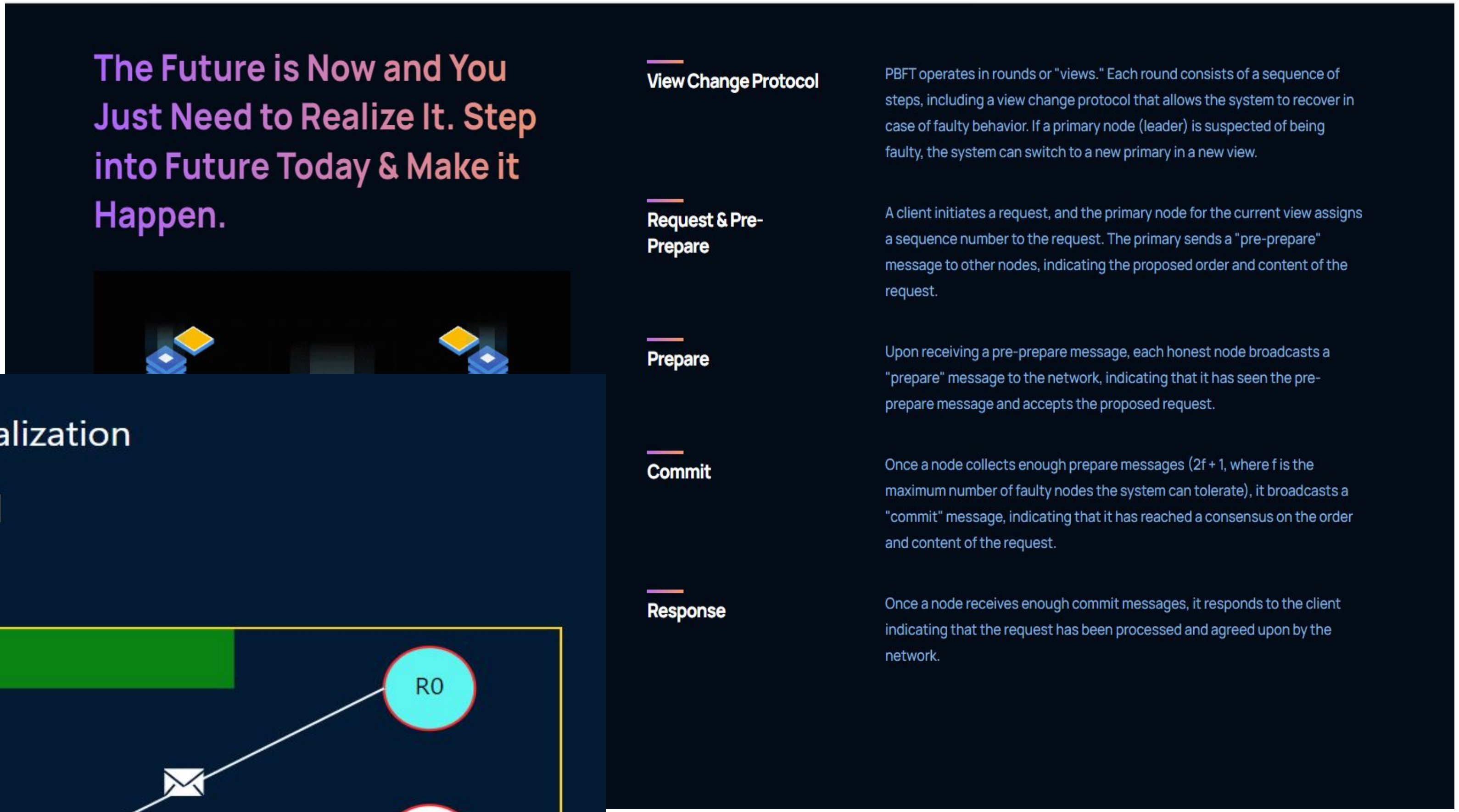
UC DAVIS COMPUTER SCIENCE

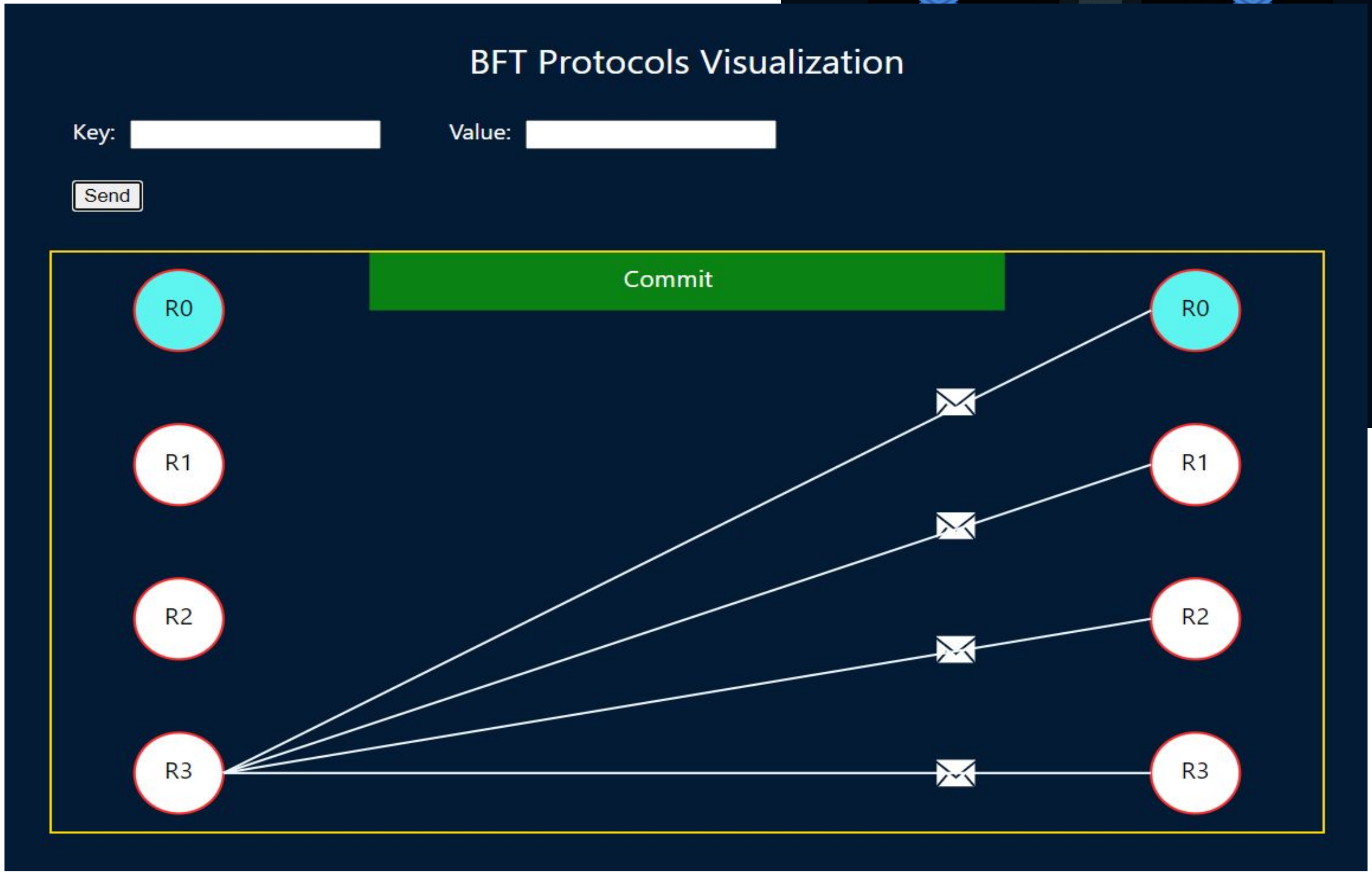
Demo





UCDAVIS
COMPUTER SCIENCE





UCDAVIS
COMPUTER SCIENCE