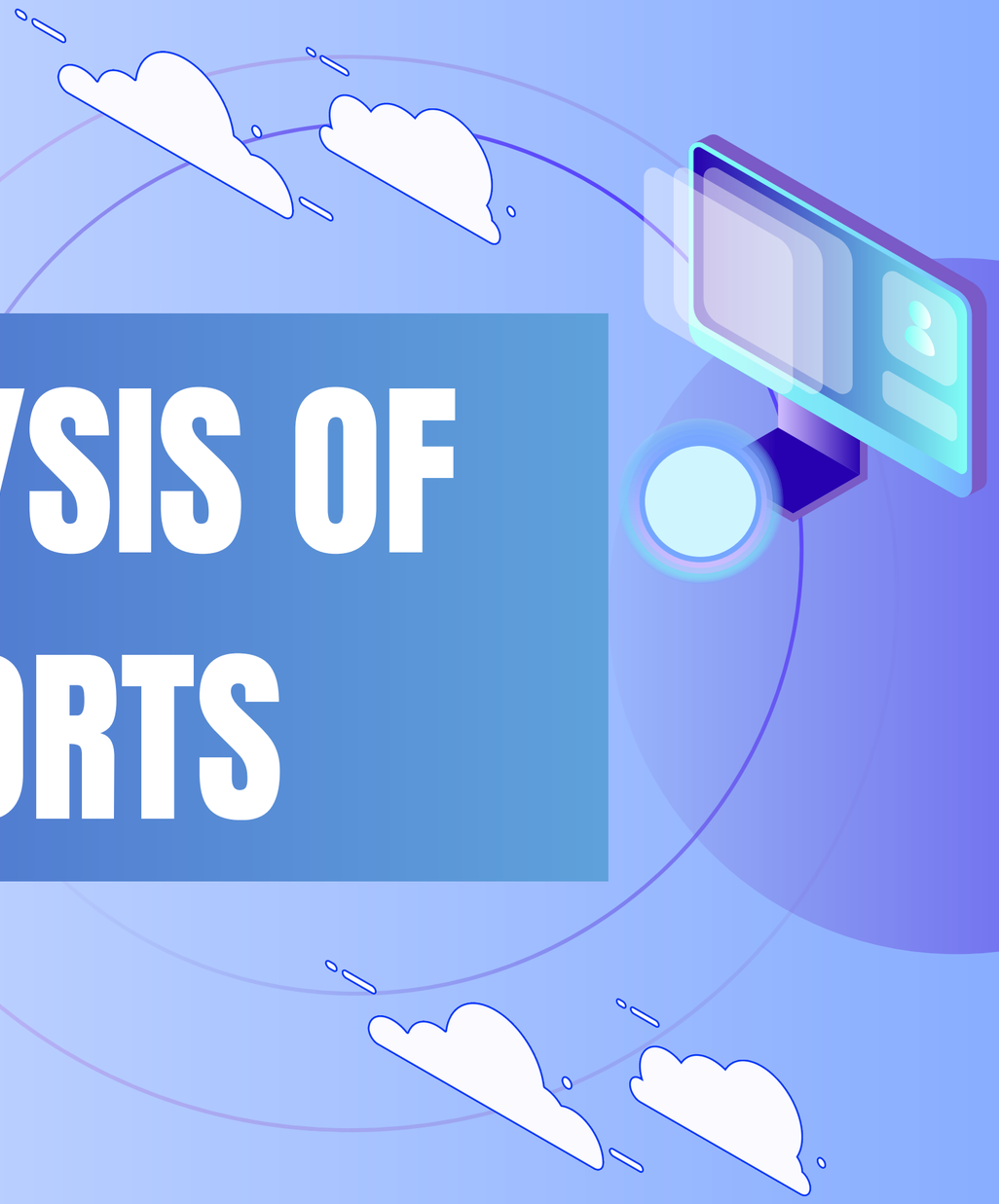
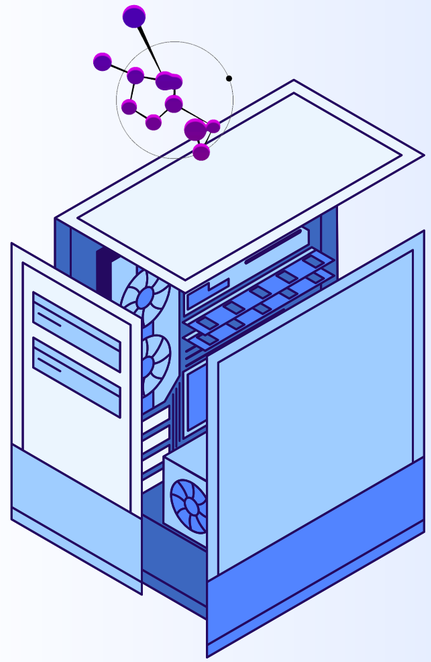
EXPLORATORY ANALYSIS OF SOFTWARE BUG REPORTS

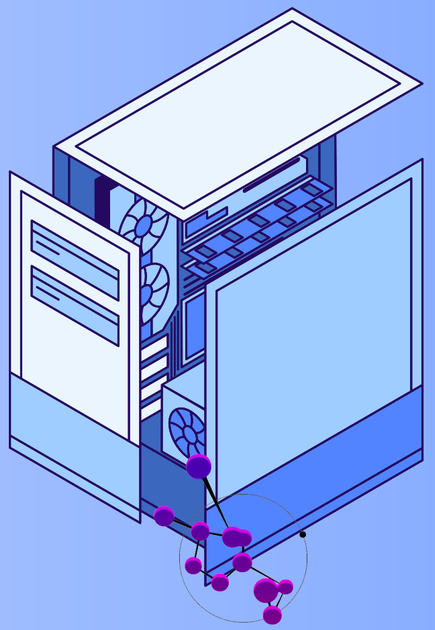


NAME: LAUREN RAPOLU
MARTICULATION NUMBER: 82065228

PROJECT OBJECTIVE



- To analyze a large-scale software bug dataset containing 50,000 bug reports
- To identify patterns in bug categories, severity levels, domains, environments, and technology stacks
- To use data visualization to extract actionable insights related to software quality and risk
- To support data-driven decision-making in software maintenance and development workflows



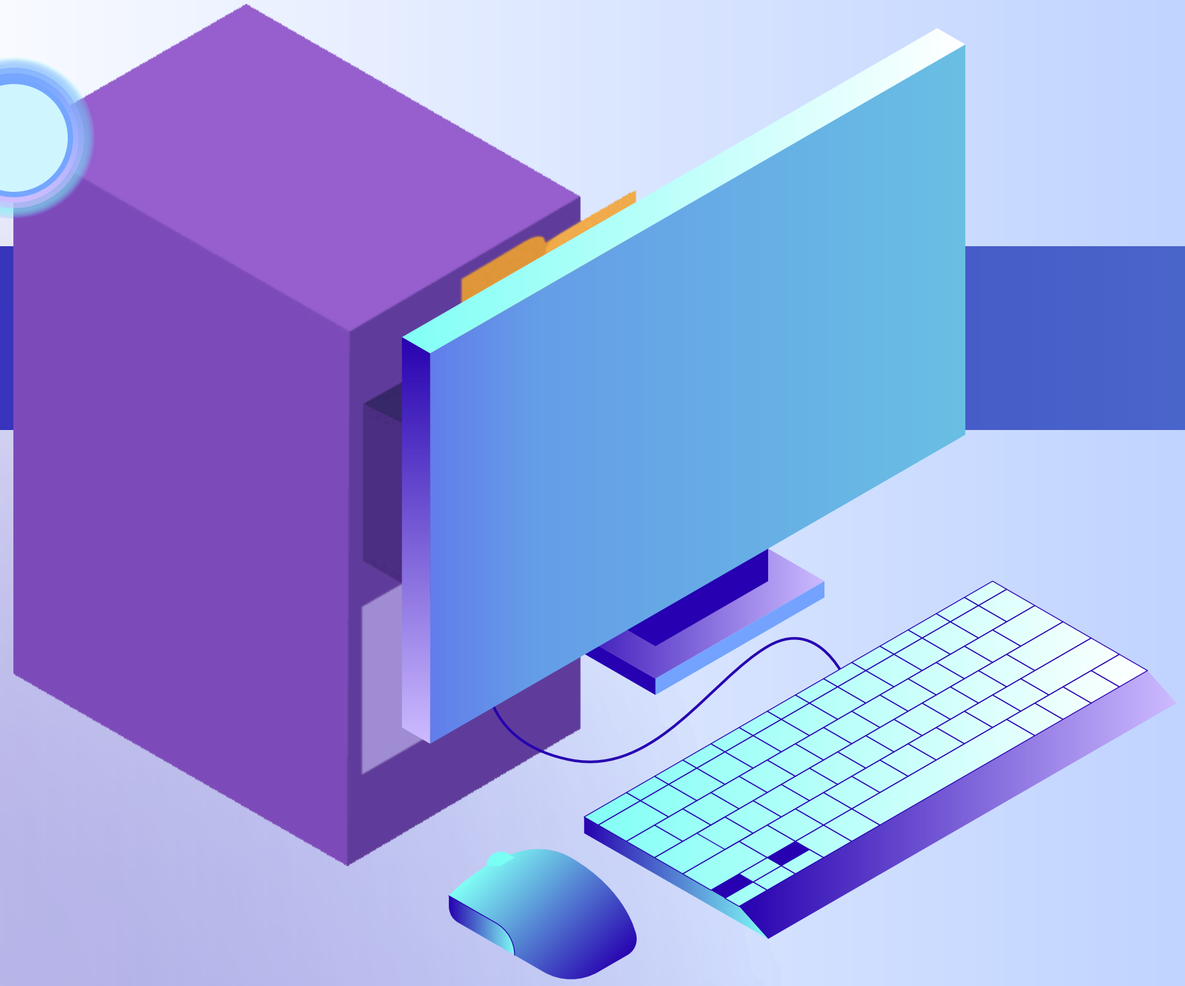
DATASET OVERVIEW

Content:

- Dataset size: 50,000 bug reports
- Each record represents a single software bug
- Data includes both textual and structured attributes

Key Features:

- Bug category, severity, domain
- Technology stack and environment
- Developer role and error codes



METHODOLOGY

- Data cleaning and preprocessing
- Standardization of categorical variables
- Exploratory Data Analysis (EDA)
- Visualization using Python (Matplotlib & Seaborn)
- Focus on understanding patterns rather than prediction

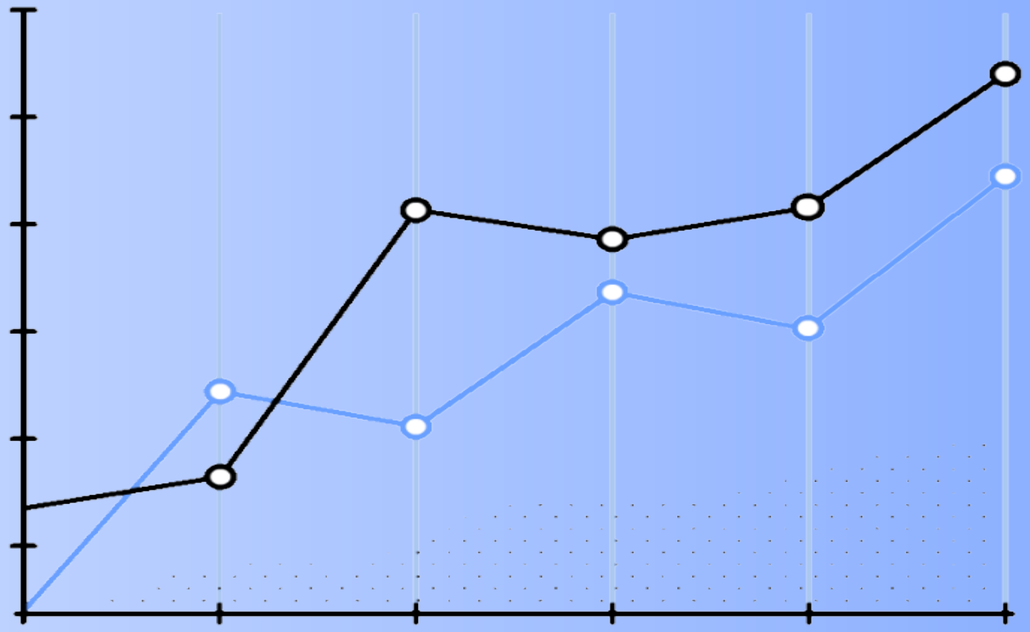
MOST COMMON BUG CATEGORIES

- Memory Leak bugs are the most frequently reported
- Logging and configuration-related bugs are also highly common
- Indicates recurring issues in resource management and system configuration

DISTRIBUTION OF

BUG SEVERITY LEVELS

- Low severity bugs occur slightly more frequently
- High and Critical bugs form a significant portion of reports
- Highlights the need for structured bug prioritization

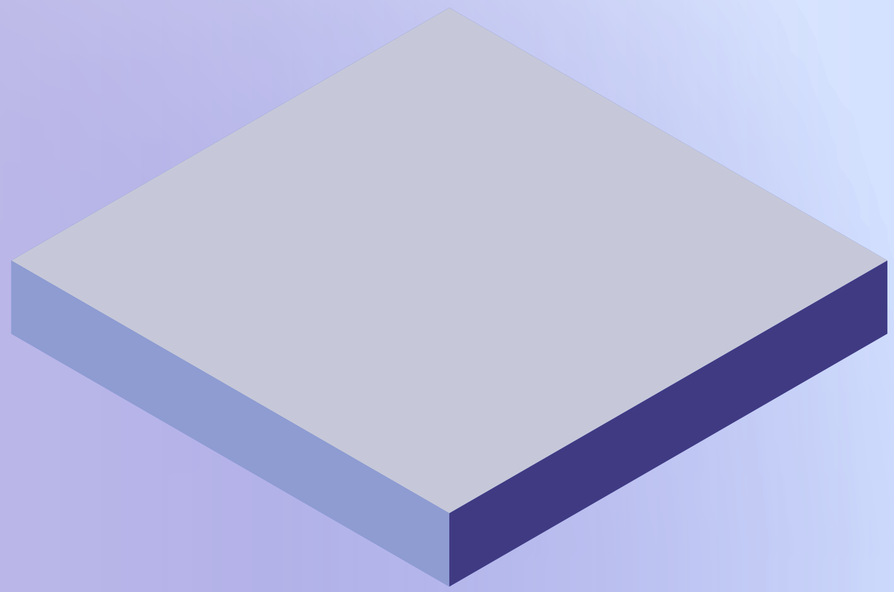


SEVERITY ACROSS DOMAINS

- Backend Systems and DevOps show marginally higher critical bug counts
- Severity distribution is fairly consistent across domains
- Core infrastructure components pose higher operational risk

HIGH-SEVERITY BUGS

BY TECHNOLOGY STACK



- Angular, MongoDB, Django, AWS, and GCP have higher severe bug counts
- Likely reflects complexity and widespread adoption
- These stacks require enhanced testing and monitoring

FUTURE TBUG SEVERITY ACROSS ENVIRONMENTS RENDS

- Most bugs are detected in Development and Staging
- Critical bugs still appear in Production
- Indicates gaps in pre-deployment testing

DEVELOPER ROLES HANDLING CRITICAL BUGS

- Security, Backend, Cloud, and Full-Stack Engineers handle most critical bugs
- High-risk issues require specialized expertise
- Highlights importance of role-based responsibility

ADDITIONAL INSIGHTS

- Certain error codes (e.g., 500, 503, 404) occur frequently
- Bug categories align strongly with developer roles
- Higher-severity bugs tend to have longer descriptions
- Specific domain–tech stack combinations act as bug hotspots

Conclusion:

- Bug data reveals clear patterns in severity, domains, and technologies
- Data visualization helps identify high-risk areas effectively

Recommendations:

- Improve testing for backend and cloud systems
- Strengthen pre-production validation
- Use bug analytics dashboards for continuous monitoring

THANK YOU!