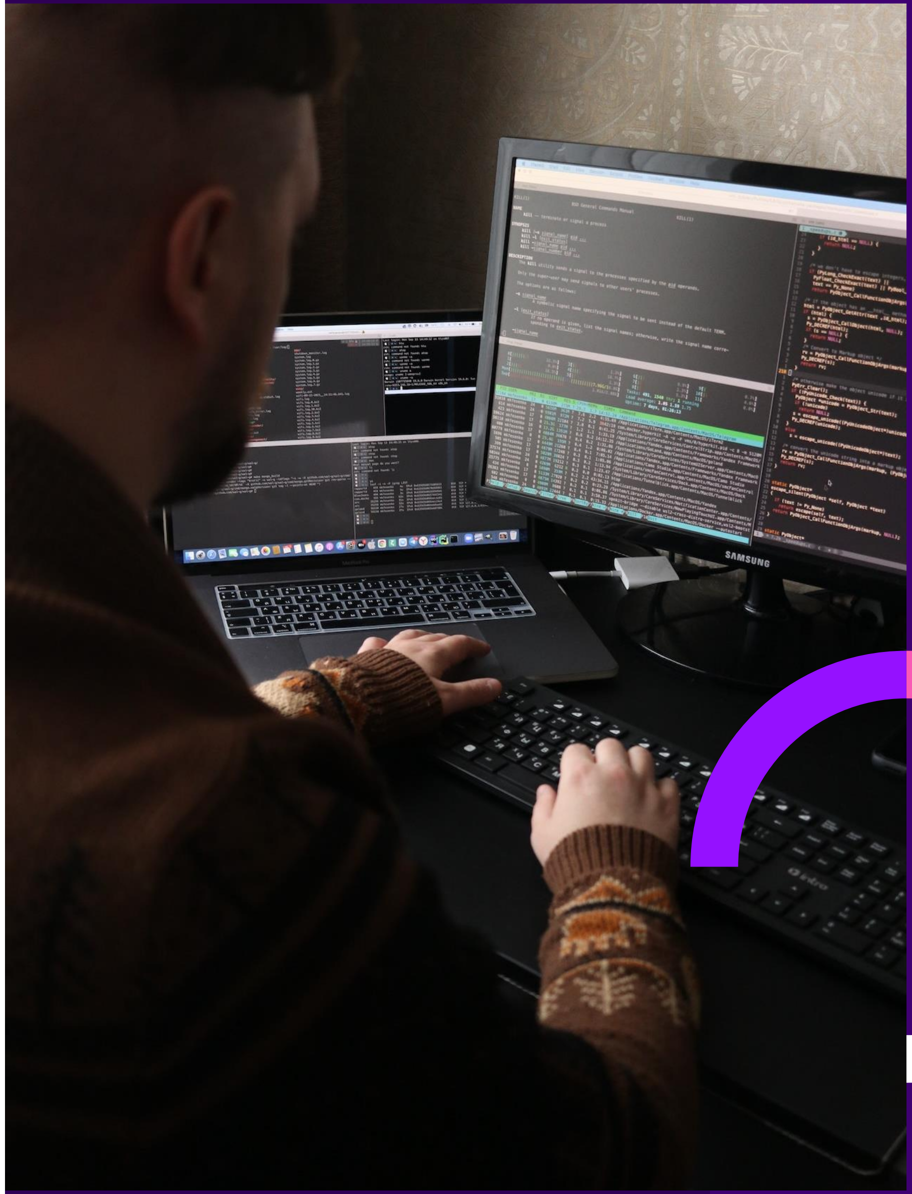
MATRIX CALCULATOR

START

C
Programming

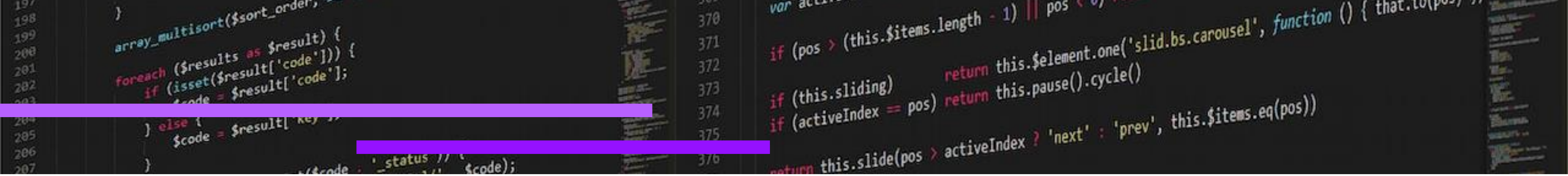
Contents

1. Motivation
2. Highlights
3. Learn from this project
4. Improvement on more time
5. Future scope
6. Contribution of each members



MOTIVATION

We chose the project of Matrix calculator because we have always been fascinated by matrices and their applications in various fields such as mathematics and data analysis. We wanted to deepen us understanding of matrices and their operations, and building a Matrix calculator seemed like the perfect way to do so. Additionally, we saw the potential for the Matrix calculator to be a useful tool for students.



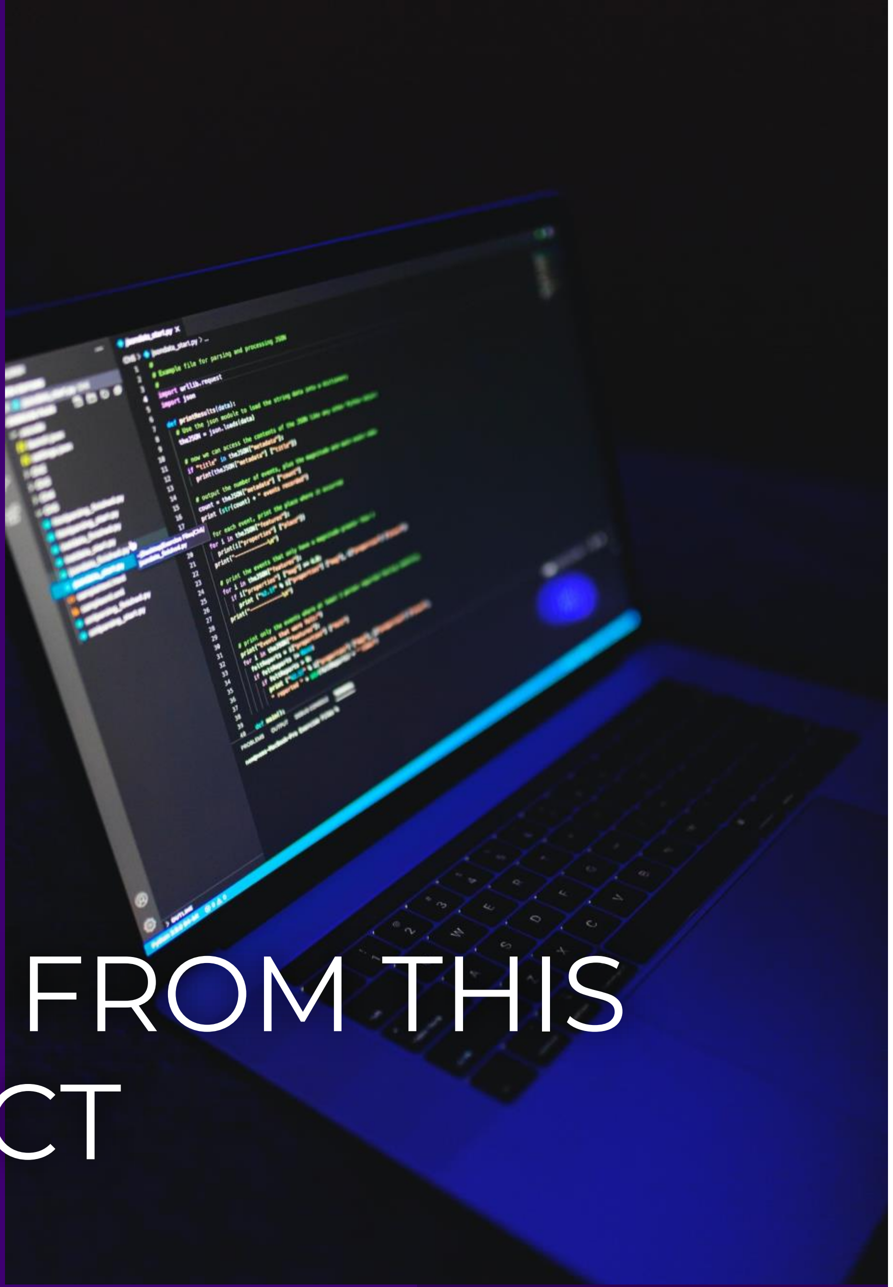
HIGHLIGHTS

a) <u>User-friendly interface</u>: The Matrix calculator project focused on creating a user-friendly interface that allows users to easily input matrices and perform various operations.

b) <u>Matrix operations</u>: The project implemented a wide range of matrix operations, including addition, subtraction, multiplication, determinant calculation, inverse calculation, scalar multiplication and transpose of the matrix.

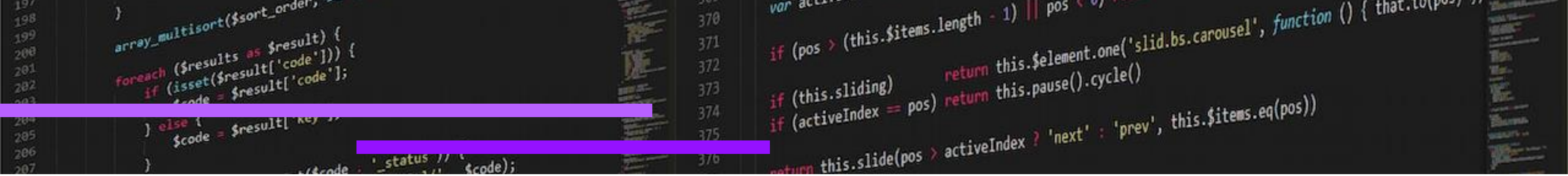
c) <u>Error handling</u>: The Matrix calculator incorporated robust error handling to ensure that users are informed about any invalid operations or input errors, helping them understand and correct their mistakes.

d) <u>Cross-platform compatibility</u>: The Matrix calculator was designed to be compatible with different operating systems and different compilers , making it accessible to a wider range of users.



LEARN FROM THIS PROJECT

Through the Matrix calculator project, I gained a deep understanding of matrix operations and their implementation. I learned how to design and develop a user-friendly interface, handle various types of user input. Moreover, I enhanced my problem-solving skills and became proficient in optimizing algorithms for better performance. Additionally, I improved my programming skills and gained experience in working on a coding projects from start to finish.



Improvement on more time

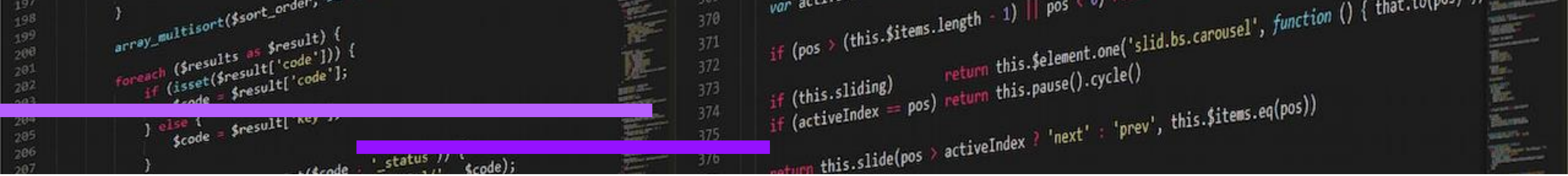
Given more time, there are several areas where I could have improved the Matrix calculator project:

a) Enhanced functionality : I could have implemented additional advanced matrix operations such as eigenvalue calculation, singular value decomposition, and matrix factorization.

b) Graphical visualization : I could have incorporated graphical visualization capabilities to help users better understand the matrix operations by representing matrices and their transformations visually.

c) Unit testing : With more time, I would have implemented comprehensive unit tests to ensure the correctness and robustness of the Matrix calculator.

d) Performance optimization : I could have further optimized the performance of matrix operations by fine-tuning the algorithms and exploring parallel computing techniques to handle larger matrices efficiently.



FUTURE SCOPE

The Matrix calculator project has a promising future with several potential areas of expansion and improvement:

a) Additional matrix operations : The Matrix calculator can be expanded to include more advanced operations such as finding rank & nullity of matrices, converting to echelon form, & various others.

b) Integration with other tools : The Matrix calculator can be integrated with other mathematical tools or software frameworks to provide a seamless environment for performing complex mathematical computations involving matrices.

c) Mobile application : Developing a mobile application version of the Matrix calculator would enable users to access and utilize the functionality of the calculator on their smartphones and tablets.

d) Support for complex numbers : Adding support for complex matrices and complex number operations would enhance the versatility of the Matrix calculator, expanding its usability in fields like quantum mechanics and signal processing.

CONTRIBUTION OF EACH MEMBERS

1. Rohan Gajendra Rudra (22EE01042):

- Designed the overall architecture of the Matrix calculator software.
- Implement Inverse, Adjoint and Symmetric Function.

2. Rahit Mondal (22MM02011):

- Implement Determinant and Cofactor Function
- Make the README in GitHub.
- Make this presentation.

3. Priyanshu Mall (22EE01041):

- Implement Addition, Subtraction and Scaler Function.
- Presentation Research.

4. Shamba Banerjee (22EE01006):

- Implement Rank, Swap Function and others research.

5. Pavan Mahadeo Wankhade (22EE01040):

- Implement Matrix Multiplication and Transpose Function.

work
blog
about
contact
My name is Lee Campbell & I'm a multi-disciplinary UX & UI Designer.
WORK
PROJECT A

Thank You