Lab1

Linear Programming Solver

Team Members

عمر طاهر حسين زيدان	22010966
يوسف احمد عبد الغني حسب الله	22011369
عبدالرحمن السيد سعد سليمان نوار	22010869

Linear Programming Solver

Class Constructor

`__init__` Method

The constructor initializes the solver with:

- `objective`: The coefficients of the objective function.
- `constraints`: The coefficients of constraint equations.
- `rhs`: The right-hand side values of constraints.
- `constraint_types`: The types of constraints (`<=`, `>=`, `=`).
- `var_restrictions`: Restrictions on variables.
- `method`: The solving method (`simplex`,`big-m`, `two-phase`, etc.).
- `type`: The optimization type (`max` or `min`).

The objective function is negated if the problem type is minimization.

Solving Methods

`solve` Method

This method determines which solving algorithm to apply based on the `method` attribute.

Simplex Solver

The Simplex Solver is designed to solve linear programming problems using the Simplex Method. The key components of this implementation are:

1.Initialization

The solver takes as input:

- A matrix of constraint coefficients.
- A vector of constraint right-hand side (RHS) values.
- A vector of objective function coefficients.
- A binary vector indicating unrestricted variables.
- A boolean flag indicating whether the problem is a maximization or minimization problem.

These inputs are converted into a structured format suitable for the Simplex algorithm.

2.Handling Slack and Unrestricted Variables

- Slack variables are added to convert inequalities into equalities.
- Unrestricted variables are split into the difference of two non-negative variables.

3.Simplex Algorithm Execution

- Identify the entering variable based on the most negative coefficient in the objective function.
- Compute the minimum ratio test to determine the leaving variable.
- Perform row operations to update the tableau.
- Repeat the process until an optimal solution is reached or an unbounded solution is detected.

big_m_method

The Big M Method incorporates artificial variables into the objective function with a large penalty coefficient (M):

• Artificial variables are added for equality (=) and ≥ constraints.

• The objective function is modified by penalizing artificial variables with a large negative or positive M.

• The Simplex algorithm proceeds normally, ensuring artificial variables leave the basis as soon as possible.

If artificial variables remain in the final solution, the problem is infeasible.

two_phase_method

The Two-Phase Method avoids the use of a large M by solving the problem in two steps:

Phase 1: Feasibility Check

• A temporary objective function is defined as the sum of all artificial variables.

• The Simplex algorithm is applied to minimize this function.

• If the optimal value is nonzero, the original problem is infeasible.

Phase 2: Standard Simplex Execution

• The original objective function is restored, and artificial variables are removed.

• The Simplex method continues to find the optimal solution.

Preemptive Goal Programming

Preemptive Goal Programming is an extension of linear programming that prioritizes multiple goals by minimizing deviations from their target values in a hierarchical manner.

1.Initialization

The solver initializes with:

- A matrix of goal coefficients and RHS values.
- Constraint coefficients and values.
- A binary vector for unrestricted variables.
- Goal directions (>=, <=, or ==).

2.Constructing the Initial Tableau

- Deviation variables (positive and negative) are introduced to account for goal violations.
- Structural constraints are added as part of the standard linear programming model.

3.Handling Unrestricted Variables

Similar to the Simplex Solver, unrestricted variables are represented as the difference between two non-negative variables.

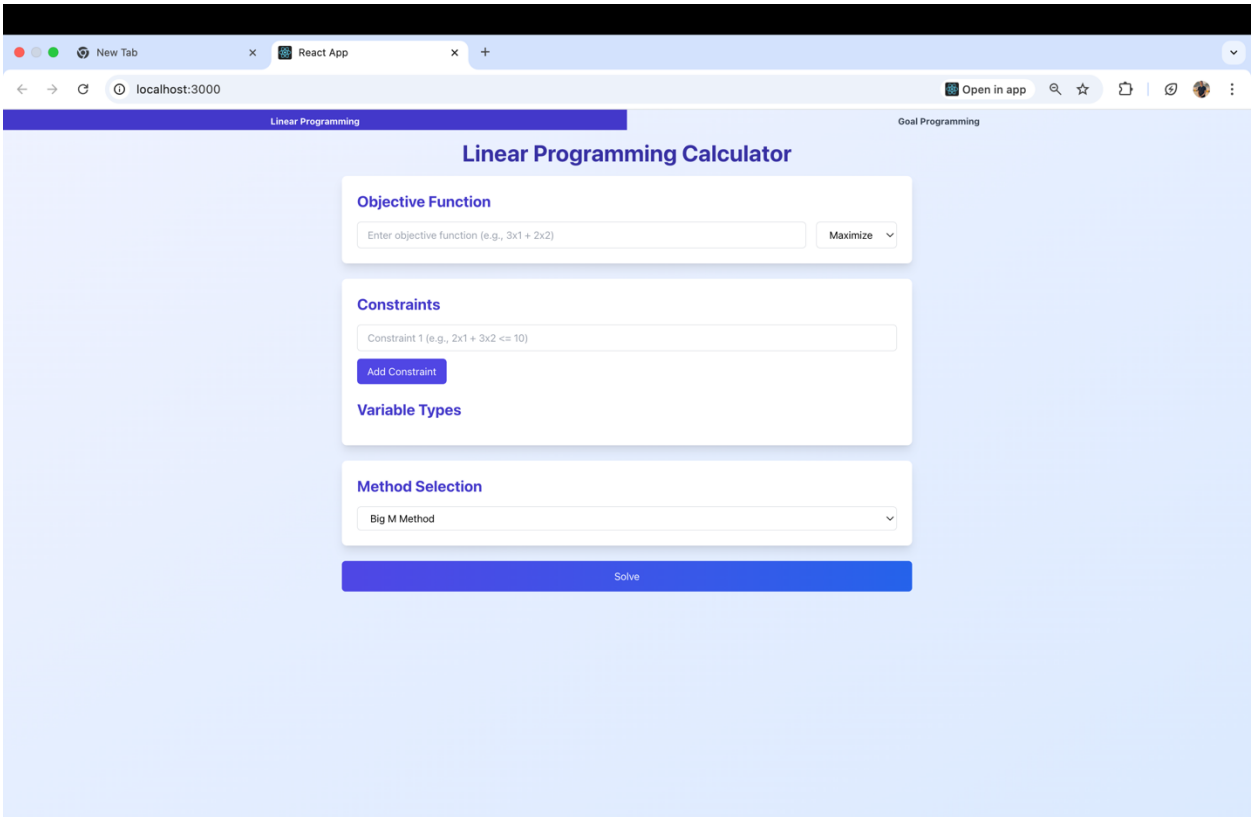
4.Objective Function Setup

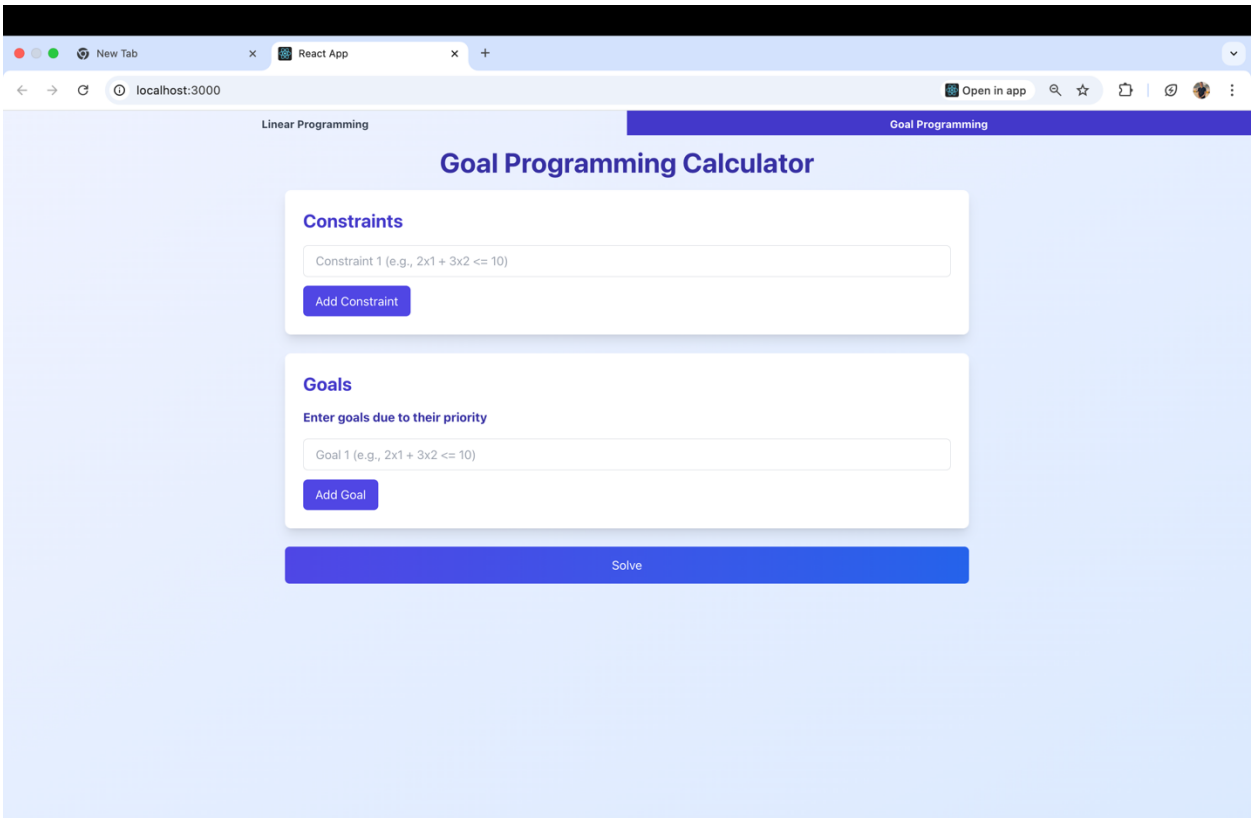
Each priority level has a separate objective function that minimizes deviations from its corresponding goal. The objectives are processed sequentially, ensuring higher-priority goals are optimized first.

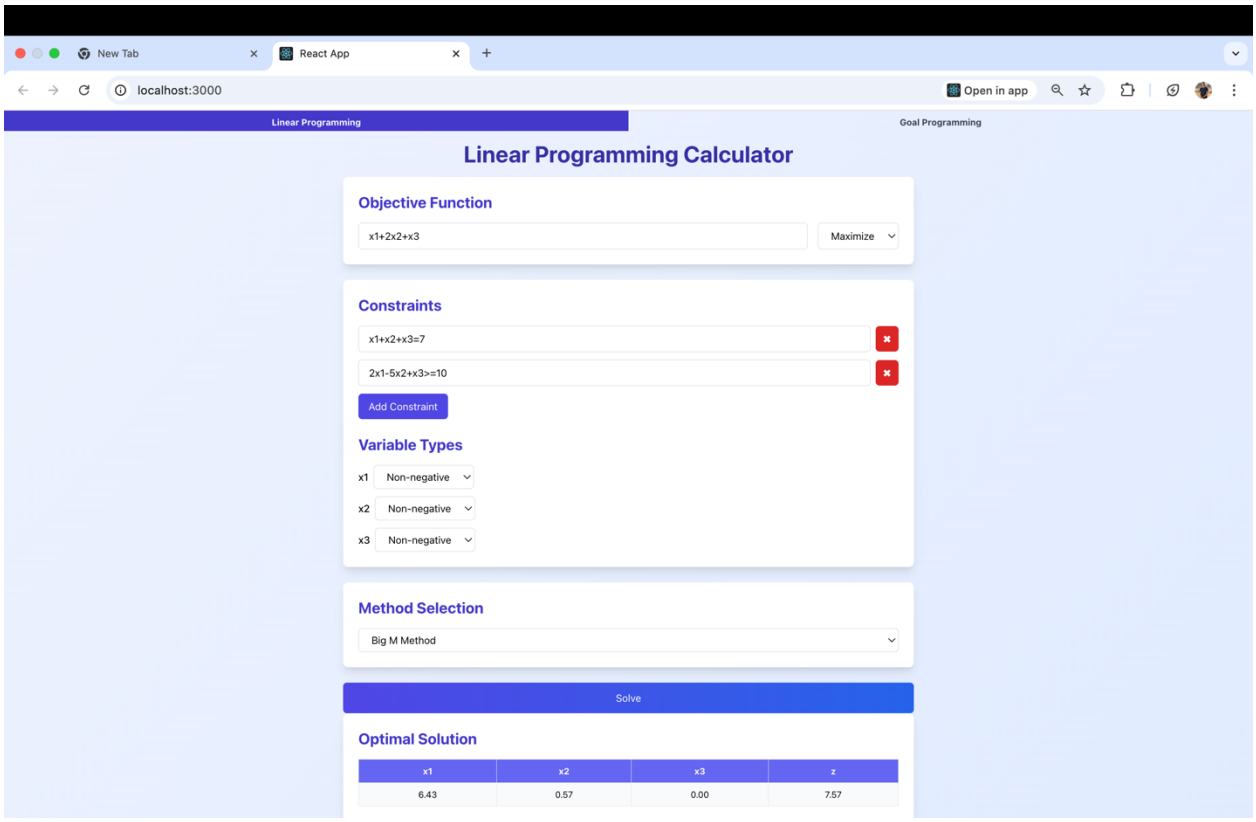
5.Solution Process

- The tableau is updated iteratively.
- Basic variables are adjusted based on pivoting operations.
- The final solution provides the optimized values for the decision variables and deviation measures.

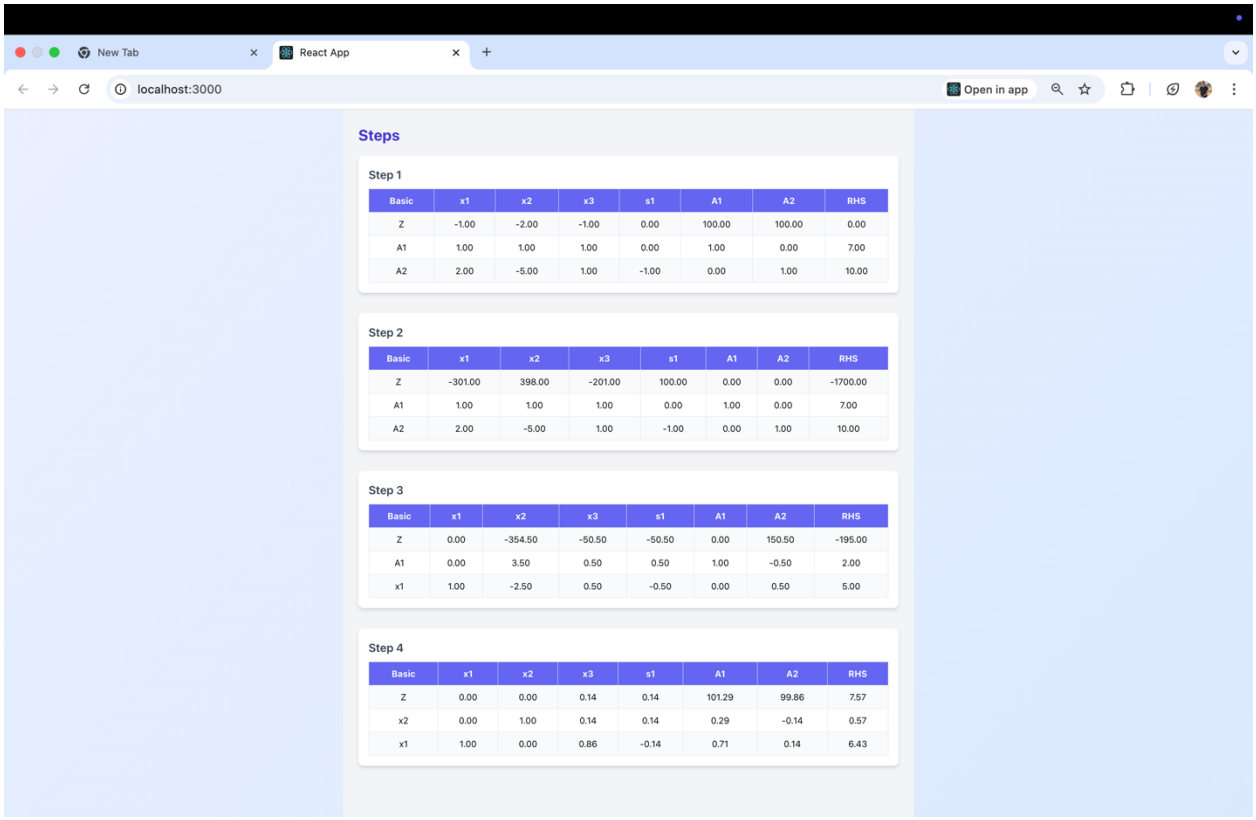
ScreenShots from application



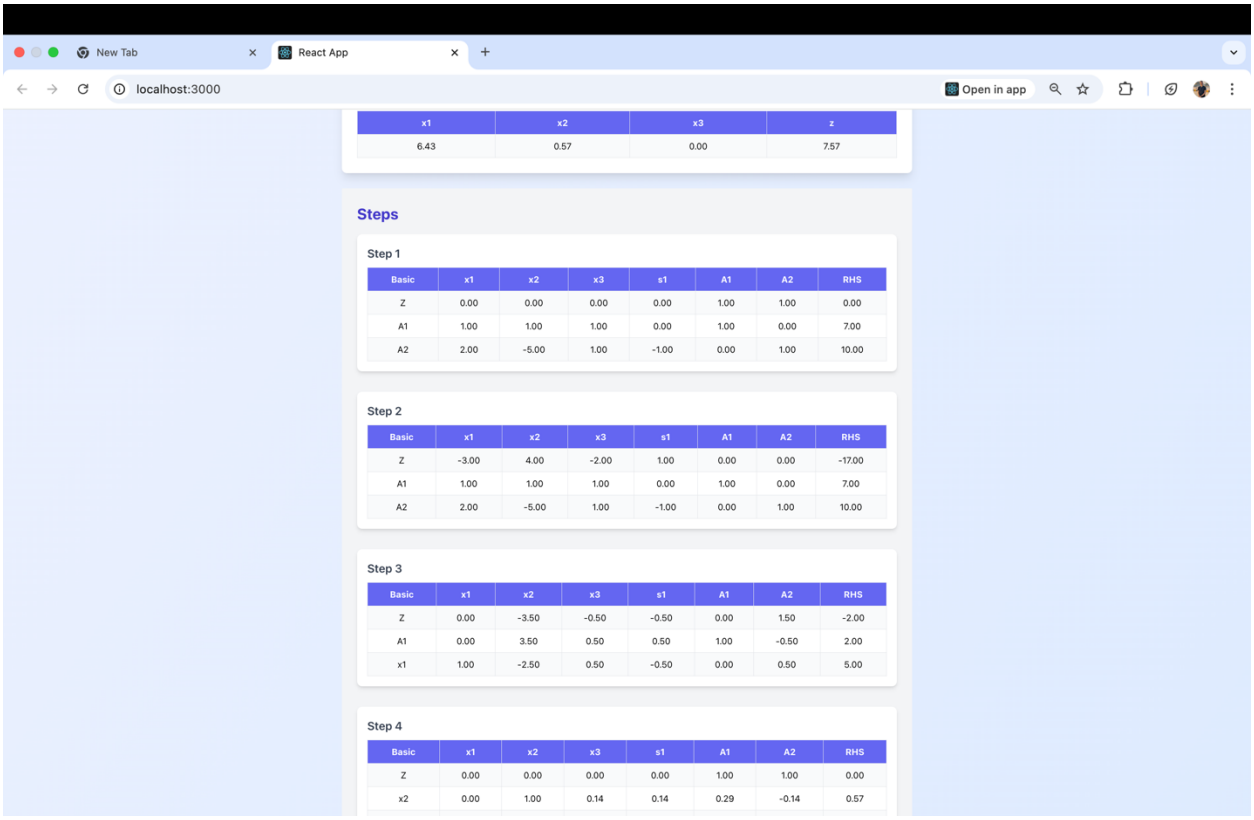



New Tab
React App
localhost:3000
Open in app
Linear Programming
Goal Programming
Linear Programming Calculator
Objective Function
x1+2x2+x3
Maximize
Constraints
x1+x2+x3=7
2x1-5x2+x3>=10
Add Constraint
Variable Types
x1 Non-negative
x2 Non-negative
x3 Non-negative
Method Selection
Big M Method
Solve
Optimal Solution
x1 x2 x3 z
6.43 0.57 0.00 7.57

Big m steps


New Tab
React App
localhost:3000
Open in app
Steps
Step 1
Basic x1 x2 x3 s1 A1 A2 RHS
Z -1.00 -2.00 -1.00 0.00 100.00 100.00 0.00
A1 1.00 1.00 1.00 0.00 1.00 0.00 7.00
A2 2.00 -5.00 1.00 -1.00 0.00 1.00 10.00
Step 2
Basic x1 x2 x3 s1 A1 A2 RHS
Z -301.00 398.00 -201.00 100.00 0.00 0.00 -1700.00
A1 1.00 1.00 1.00 0.00 1.00 0.00 7.00
A2 2.00 -5.00 1.00 -1.00 0.00 1.00 10.00
Step 3
Basic x1 x2 x3 s1 A1 A2 RHS
Z 0.00 -354.50 -50.50 -50.50 0.00 150.50 -195.00
A1 0.00 3.50 0.50 0.50 1.00 -0.50 2.00
x1 1.00 -2.50 0.50 -0.50 0.00 0.50 5.00
Step 4
Basic x1 x2 x3 s1 A1 A2 RHS
Z 0.00 0.00 0.14 0.14 101.29 99.86 7.57
x2 0.00 1.00 0.14 0.14 0.29 -0.14 0.57
x1 1.00 0.00 0.86 -0.14 0.71 0.14 6.43

Two-phase steps



x1	x2	x3	z
6.43	0.57	0.00	7.57

Steps

Step 1

Basic	x1	x2	x3	s1	A1	A2	RHS
Z	0.00	0.00	0.00	0.00	1.00	1.00	0.00
A1	1.00	1.00	1.00	0.00	1.00	0.00	7.00
A2	2.00	-5.00	1.00	-1.00	0.00	1.00	10.00

Step 2

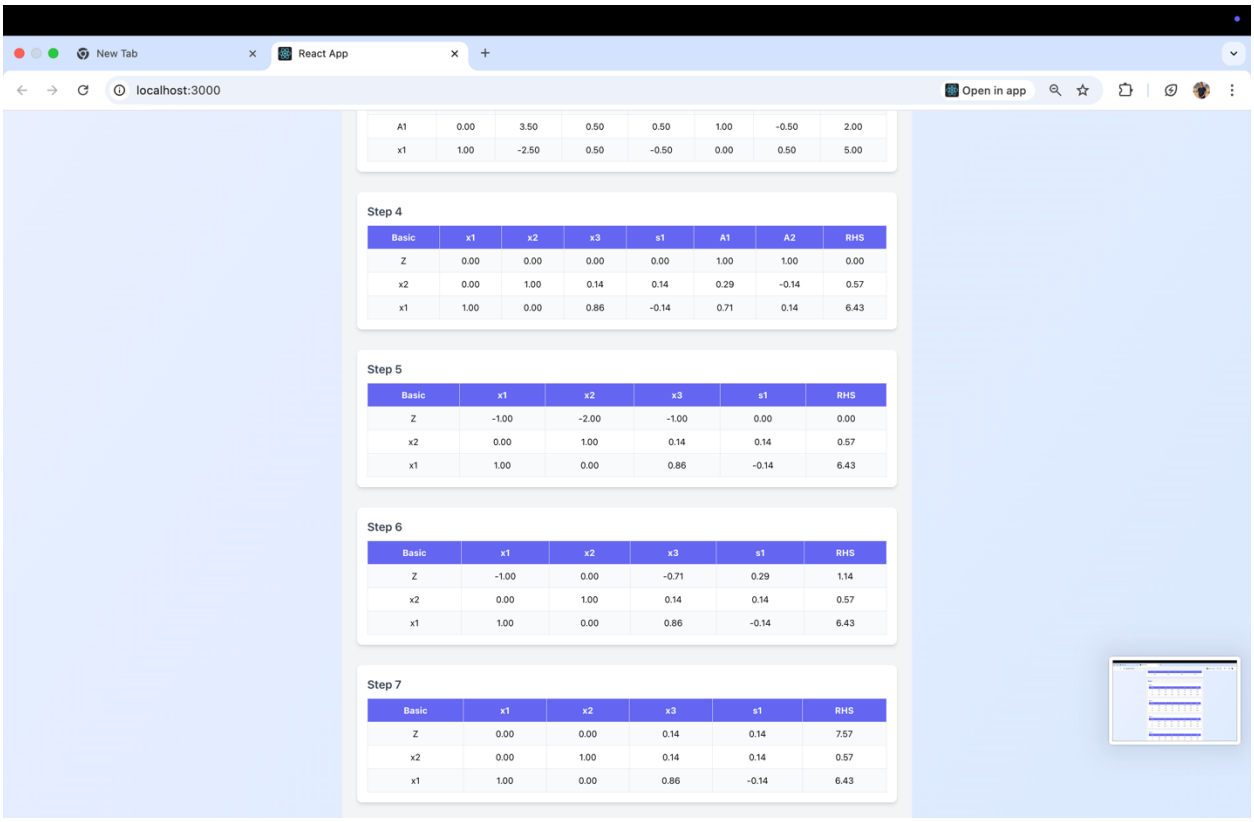
Basic	x1	x2	x3	s1	A1	A2	RHS
Z	-3.00	4.00	-2.00	1.00	0.00	0.00	-17.00
A1	1.00	1.00	1.00	0.00	1.00	0.00	7.00
A2	2.00	-5.00	1.00	-1.00	0.00	1.00	10.00

Step 3

Basic	x1	x2	x3	s1	A1	A2	RHS
Z	0.00	-3.50	-0.50	-0.50	0.00	1.50	-2.00
A1	0.00	3.50	0.50	0.50	1.00	-0.50	2.00
x1	1.00	-2.50	0.50	-0.50	0.00	0.50	5.00

Step 4

Basic	x1	x2	x3	s1	A1	A2	RHS
Z	0.00	0.00	0.00	0.00	1.00	1.00	0.00
x2	0.00	1.00	0.14	0.14	0.29	-0.14	0.57



A1	0.00	3.50	0.50	0.50	1.00	-0.50	2.00
x1	1.00	-2.50	0.50	-0.50	0.00	0.50	5.00

Step 4

Basic	x1	x2	x3	s1	A1	A2	RHS
Z	0.00	0.00	0.00	0.00	1.00	1.00	0.00
x2	0.00	1.00	0.14	0.14	0.29	-0.14	0.57
x1	1.00	0.00	0.86	-0.14	0.71	0.14	6.43

Step 5

Basic	x1	x2	x3	s1	RHS
Z	-1.00	-2.00	-1.00	0.00	0.00
x2	0.00	1.00	0.14	0.14	0.57
x1	1.00	0.00	0.86	-0.14	6.43

Step 6

Basic	x1	x2	x3	s1	RHS
Z	-1.00	0.00	-0.71	0.29	1.14
x2	0.00	1.00	0.14	0.14	0.57
x1	1.00	0.00	0.86	-0.14	6.43

Step 7

Basic	x1	x2	x3	s1	RHS
Z	0.00	0.00	0.14	0.14	7.57
x2	0.00	1.00	0.14	0.14	0.57
x1	1.00	0.00	0.86	-0.14	6.43

Goal programming