

# Railway Deadlock Simulation Using Banker's Algorithm

Simulating railway deadlocks to ensure safe track management.

By Yash Amte, Akshay Pise, Rohit Bharti, Prathamesh Dhomne



# Introduction to Deadlock in Railways



## Deadlock Concept

Blocking where trains wait indefinitely for tracks



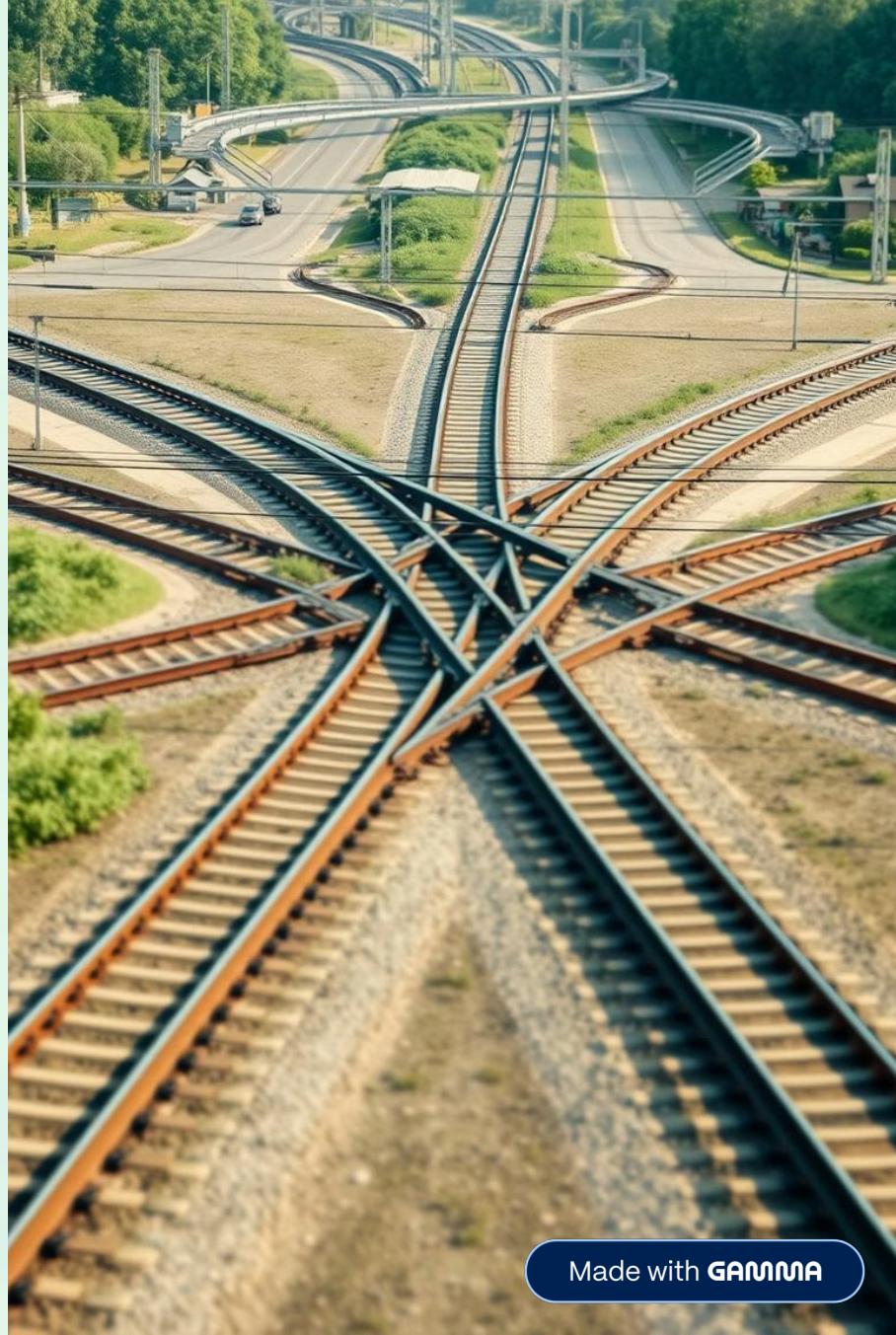
## Causes

Multiple trains requiring shared tracks simultaneously



## Impact

Delays, safety hazards, inefficiencies



# Banker's Algorithm Overview

## Purpose

Detect and avoid deadlocks by checking safe states

## Core Idea

Ensure each resource allocation keeps system safe

## Application

Allocating tracks without causing deadlocks

```

resource105 (
resource allocio);
▼ ryctice alleartfaloviecid);
▼ safetly ieale);
safalrn'by sterecl resource (anoplay)>
  ▷ taain'16 sester (eccon(agetc));
    reerca'*=11;;
  ▷ safin''86-slatigroial;
  ▷ wiple contaion teally>
  ▷ viit tenturational ertviel. (eatst);
  ▷ wiple a Battentfirgeifal (tenerity)
  ▷ taain''resation decatal);
  ▷ night-tyye locsion legett);
  ▷ wiple 8 Battelci ercaiere);
  ▷ train-82 sartom exgo);
  ▷ wiple 8 Batter tagginster (aget);
  ▷ "fly's-pedeolfal faszim_lapotic);
  ▷ wight.cecentral tecaly);
  ▷ "tadin"(teafic (rectc));
  ▷ winit 8 Battent ancaltarl);
  ▷ will'nsaide (zages);
  ▷ wiple Behientral angurrent.(seget);
  ▷ ponestr;(saetc));
  ▷ desyinslock());
  "("; aparcate, Tee--ad);
  ▷ cealcy());
  ▷ wight Sable Berter, seccrty;(agec);
  ▷ eaally cale (bath));
  ▷ ctyls-seaglolid orcert);
  ▷ "teptatrect.taply);
  ▷ taain'16 scatail arcoudering (rectt);
  ▷ "tadin"(forelace, 'Taver_grest));
  ▷ wiple 8 Batter lncents(rzally);
  ▷ wiple a Batter fallly (ecomt);
  ▷ "taain"(saatiyos,out, Varqyeail);
  ▷ Wolie mrestsd);
  ▷ "mowes""tastal, cacarid = 'Oxieric);
  ▷ salyl - decadiocel inceie);
  ▷ gervles"catlock areptercl);
  ▷ statascivision,
  ▷ aylt palacr cosnderation cowtd"(capn);
  ▷ taain"cection acortci, cater_grest);
  ▷ torts"seccle perton);
  ▷ upciplecaite location our tattial;
  ▷ derface resgepavol;
  ▷ "nsterade currie, locatrat (edent
  bases "raledyins, ancougston bet
  resoning doben, incasial proce
  paties to saes parvngly"
  ▷

```

# System Design & Implementation

## C Language

Core logic of Banker's algorithm  
for track allocation

## Python HTTP Server

Backend CGI server handles  
user input requests

## HTML & CSS

Frontend interface for train and track input

# User Interaction and Input

## Input Parameters

- Number of trains
- Number of tracks
- Track allocation requests

## Process

User submits data to server for simulation

# Simulation Output & Results



## Safe State

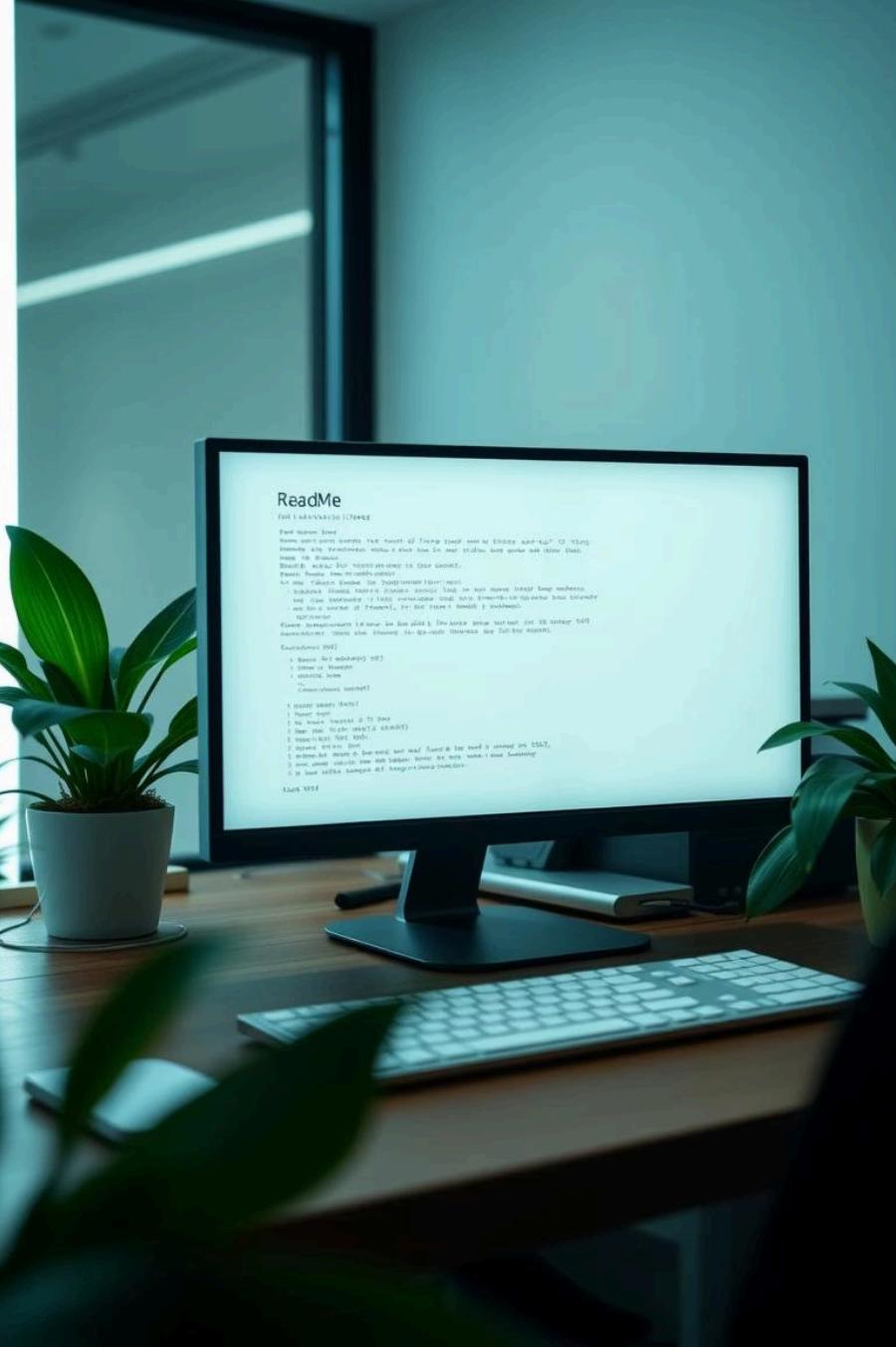
Tracks allocated without causing deadlock



## Unsafe State

Potential deadlock detected, requires action





# Additional Functionalities

## README File

Detailed setup and usage instructions

## MIT License

Open-source permissions and legal terms

## Extra Features

Extensions for better usability and documentation



# Conclusion & Future Work

## Validate Safe Operations

Ensure no deadlocks in real-time railway control

## Enhance Interface

Improve frontend usability and visualization

## Extend Algorithm

Adapt for dynamic track and train scenarios

Improving railway deadlock prevention for safe, efficient transport systems.