

# Full Stack Engineering Challenge: J241

## Cell Growth Simulation

### Background

Understanding the growth patterns of bacterial colonies is important in fields such as microbiology, medicine, and environmental science. In this challenge, you will simulate these growth patterns using web technologies. By using React and TypeScript, you'll create an interactive simulation that demonstrates how bacterial cells divide and spread within a confined space.

### Objective

Design and implement a React application that simulates the growth of a bacterial colony based on specific biological rules. The application should be capable of starting, pausing, and resetting the simulation, as well as adjusting the growth interval.

### Requirements

#### Functional Requirements

1. **Grid Representation:** Display a grid representing a petri dish (e.g., 20x20 cells) where each cell in the grid can be either empty or occupied by a bacterial cell.
2. **Growth Rules:**
  - Cells divide every fixed time interval (e.g., every 1 second).
  - Cells can only divide if there is at least one adjacent empty cell (up, down, left, right).
3. **User Controls:**
  - A button to start/pause the simulation.
  - A button to reset the simulation.
  - An input field to set the time interval for cell division.
  - Allow users to manually place or remove bacterial cells on the grid by clicking on the cells.

#### Technical Requirements

1. Use React with strict TypeScript for your application, no external libraries are allowed.
2. Efficiently manage state and simulation status.
3. Make sure the application is optimized for performance.
4. Include robust error handling to manage unexpected scenarios and user interactions.

5. Deploy the final application on a hosting platform of your choice (e.g., GitHub Pages, Vercel, Netlify, etc).

## Evaluation Criteria

- **Functionality:** The simulation works as specified.
- **Optimization:** Good performance in the real world.
- **Code Quality:** Adherence to TypeScript best practices.
- **Deployment:** Successful deployment of the application.
- **Documentation Quality:** Documentation is clear, concise, and informative.

## Bonus

1. Add a feature to visualize the growth rate of the colony over time.
2. Implement a feature to dynamically change the size of the grid.
3. Make the application accessible, including features like keyboard navigation and screen reader compatibility.

## Submission Instructions

1. Submit a ZIP file containing the source code for your application.
2. Provide a link to the live hosted application.
3. Provide a link to the source code repository.
4. A README file that contains:
  - An overview of the project and its features.
  - Instructions on how to set up and run the project locally.
  - Details on the project structure and key components.
  - Any assumptions made and any additional features implemented.
  - A performance analysis/review for your application, including specific performance metrics.