# The Petri-Dish Posse Documentation

## Overview

The **Petri-Dish Posse** is a Django-based web application designed to simulate the spread and control of infectious diseases using the SEIRS (Susceptible-Exposed-Infectious-Recovered-Susceptible) model. This application allows users to explore disease dynamics under various conditions, including scenarios with and without interventions such as vaccination and treatment.

### Key Features

* **SEIRS Model Simulation**: Models the progression of diseases by dividing the population into susceptible, exposed, infectious, and recovered groups.
* **Intervention Scenarios**: Users can run simulations with various interventions, including vaccination and treatment strategies.
* **Visualization Tools**: Generates graphs and charts to visually represent the simulation results, such as infection rates and recovery trends.
* **Customizable Parameters**: Provides a range of parameters that users can adjust to model different scenarios and explore their effects on disease spread.

## Installation Guide

### System Requirements

To run "The Petri-Dish Posse," you'll need the following:

* **Python 3. x**: A version of Python compatible with Django and the other dependencies.
* **Django**: The web framework used to build the application.
* **Matplotlib**: A library for generating plots and graphs.
* **Other Dependencies**: Listed in requirements.txt and include libraries needed for the SEIRS model and data manipulation.

**GitHub Set up  
  
Cloning the repository:**

|  |
| --- |
| bash |
| git clone <https://github.com/janymuong/meningitis_sim.git>cd meningitis\_sim |

### Environment Set-Up

Using a virtual environment is crucial for managing project-specific dependencies and avoiding conflicts with system-wide packages. Here’s how to set up a virtual environment:

1. **Create and activate a virtual environment**:
   * First, ensure that **Python 3.12** is installed on your system. You can verify this by checking the Python path:

|  |
| --- |
| bash |
| **$ which python3** |

* + Install virtualenv if it's not already installed:

|  |
| --- |
| bash |
| **$ python3 -m pip install --user virtualenv** |

* + Create a new virtual environment and activate it:

|  |
| --- |
| bash |
| **$ python3 -m virtualenv --python=<path-to-python3.12> ../.dj\_sim**  **$ source ../.dj\_sim/bin/activate** |

1. Alternatively  
   You can use the make setup command defined in the [Makefile](file:///C:\Users\Hp\AppData\Roaming\Microsoft\Word\Makefile) to automate this process.
2. **Install project dependencies**:
   * After activating the virtual environment, install the necessary packages:

|  |
| --- |
| bash |
| **$ make install** |

1. **Open the project directory**:
   * Navigate to the project directory to access and edit project files:

|  |
| --- |
| bash |
| **$ cd meningitis\_sim**  **$ code .** |

### Installation Steps

1. **Set up the virtual environment**:

|  |
| --- |
| bash |
| **make setup** |

1. **Activate the virtual environment**:

|  |
| --- |
| bash |
| **source ../.dj\_sim/bin/activate** |

1. **Install the required packages**:

|  |
| --- |
| bash |
| make install |

1. **Run the Django server**:

|  |
| --- |
| bash |
| **python3 manage.py runserver** |

1. **Run database migrations**:

|  |
| --- |
| bash |
| make migrate |

### Configuration

* No additional configuration is needed beyond the installation steps. Ensure that the virtual environment is activated whenever you run the application to maintain the correct context for dependencies.

## User Guide

### Getting Started

1. **Access the Application**:
   * Open a web browser and go to <http://127.0.0.1:8000/>. This URL will load the application’s homepage.
2. **Run a Simulation**:
   * On the homepage, select the appropriate simulation form based on the type of scenario you want to model (e.g., **standard SEIRS model**, **vaccination scenario**, etc.).
   * Input the required parameters into the form. These parameters will vary depending on the simulation type.

### Detailed Usage

#### Navigation

* **Home Page**: Provides links to different simulation forms and displays the results of previous simulations.
* **Forms**: The application includes several forms for different simulation types:
  + **Normal Simulation**: For basic SEIRS simulations without interventions.
  + **Vaccine Simulation**: For simulations that include vaccination as an intervention.
  + **Age-Based Vaccine Simulation**: For simulations with age-specific vaccination strategies.
  + **Treatment Simulation**: For simulations involving treatment as an intervention.

#### Main Functions

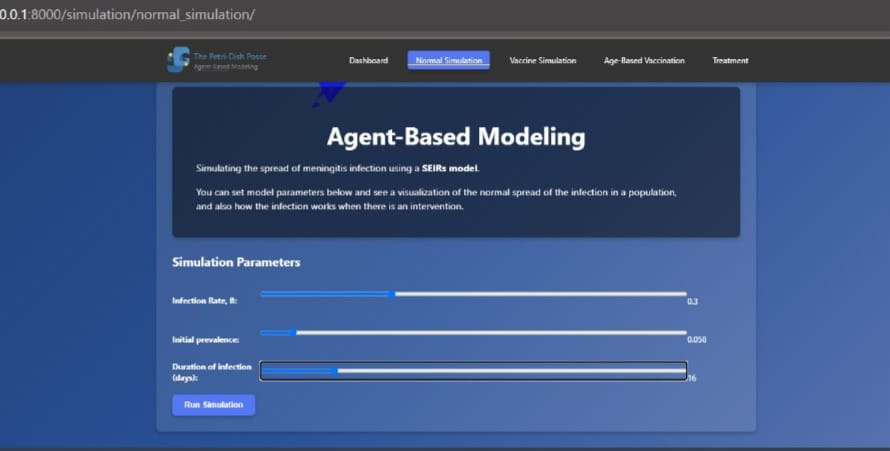
* **Run Simulation**: After setting up the simulation parameters, submit them to start the simulation. The system will process the data and generate results.
* **View Results**: Once the simulation is complete, the results will be displayed on the screen, including graphical visualizations of disease dynamics.

#### Examples and Screenshots

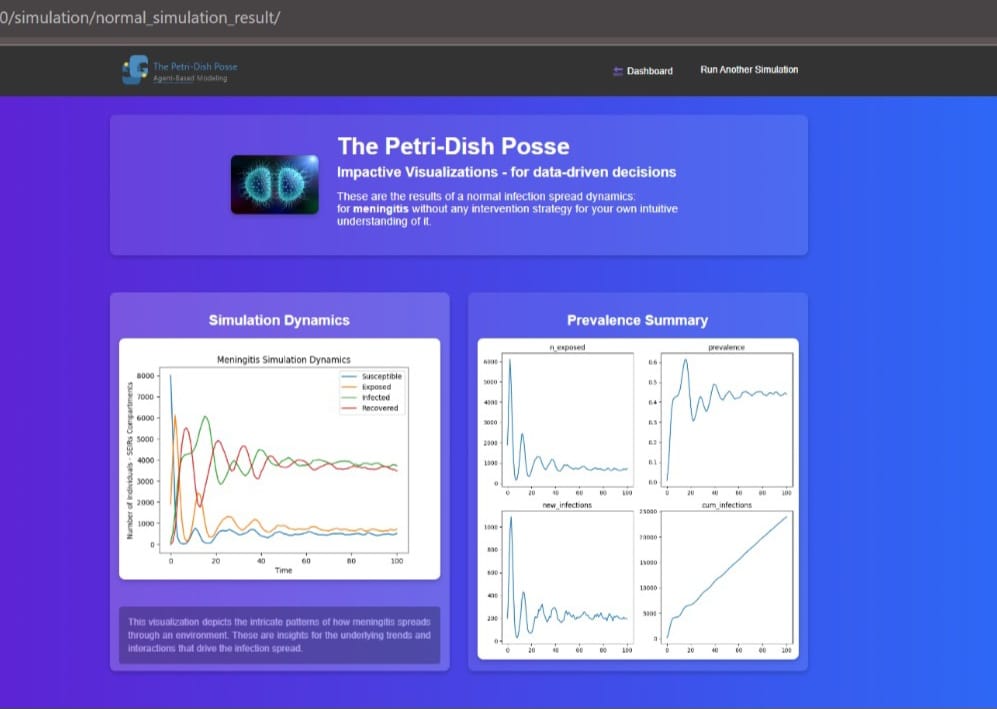
**Home Page Overview**

**Normal Simulation**

Setting up Parameters

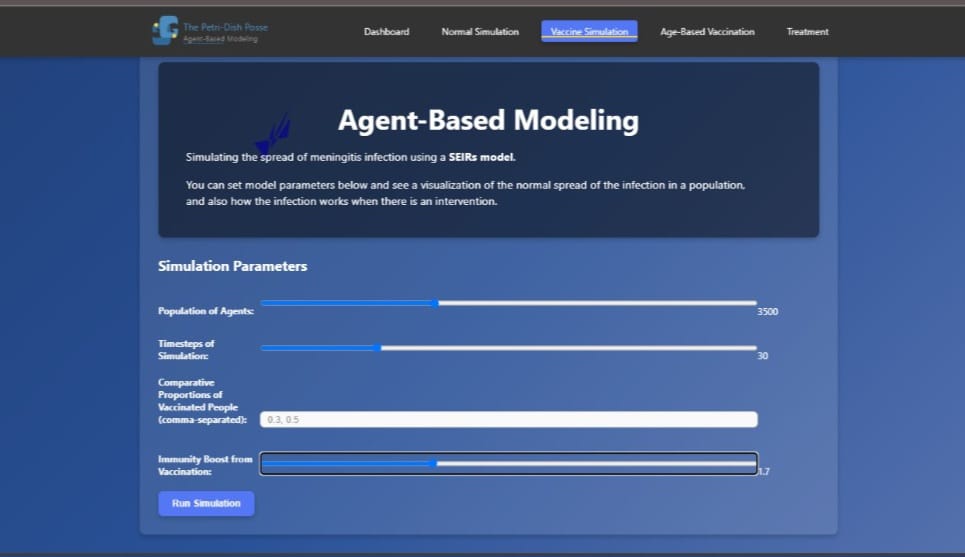
****

Visualization

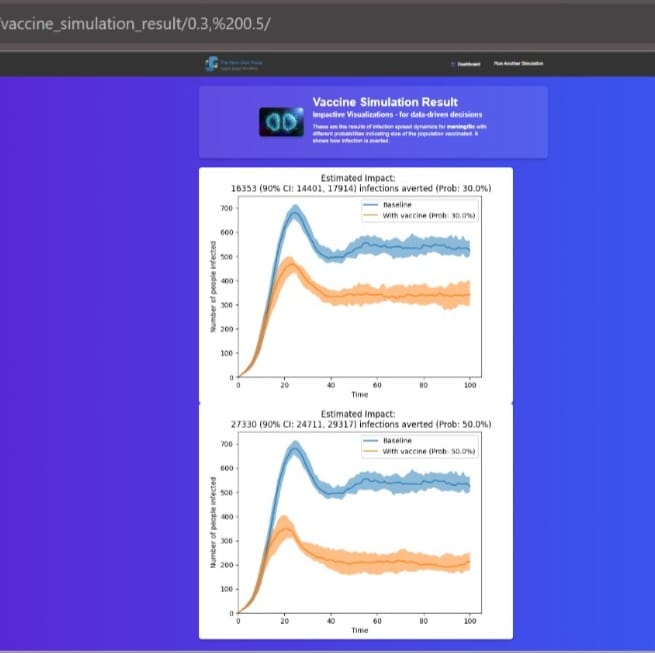
****

**Vaccine Simulation**

Setting Parameters

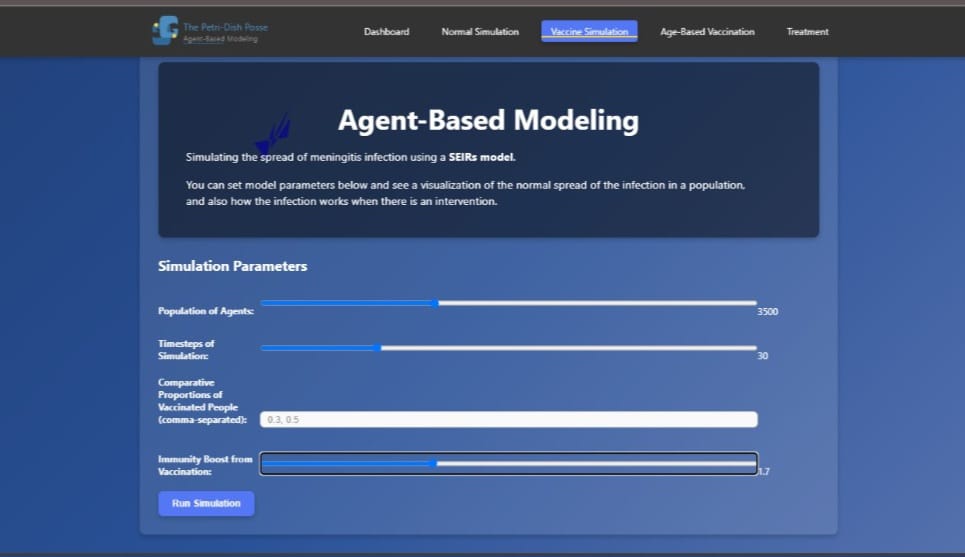


Graphical representation

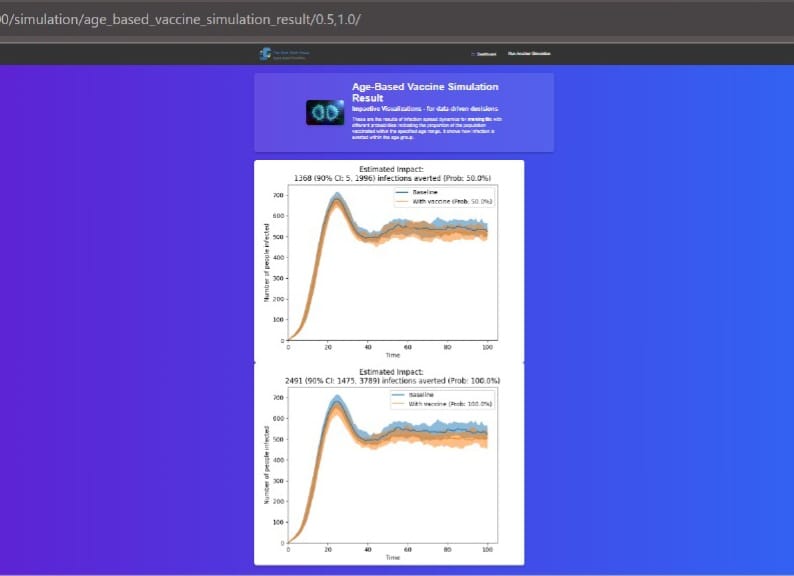


**Age-Based Vaccine Simulation**

Setting up different parameters

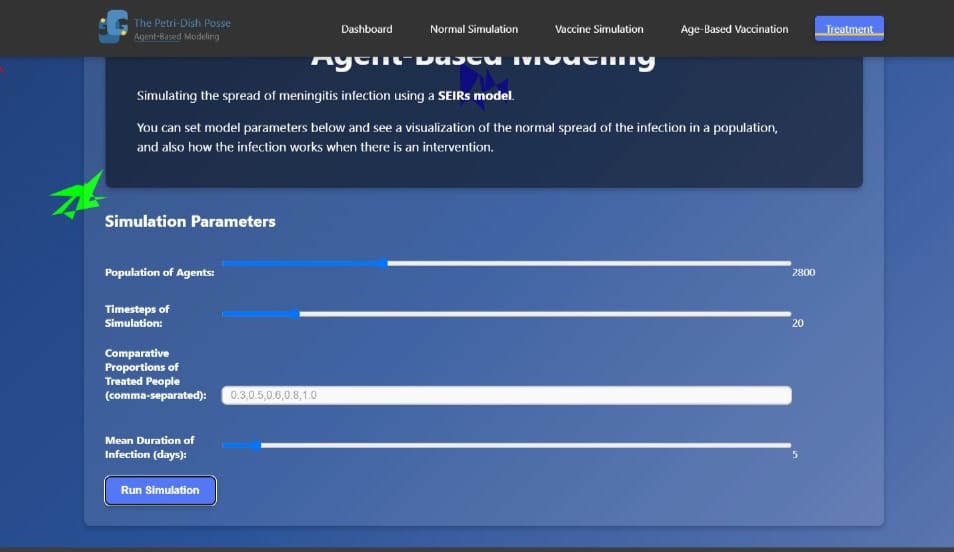


Visualization

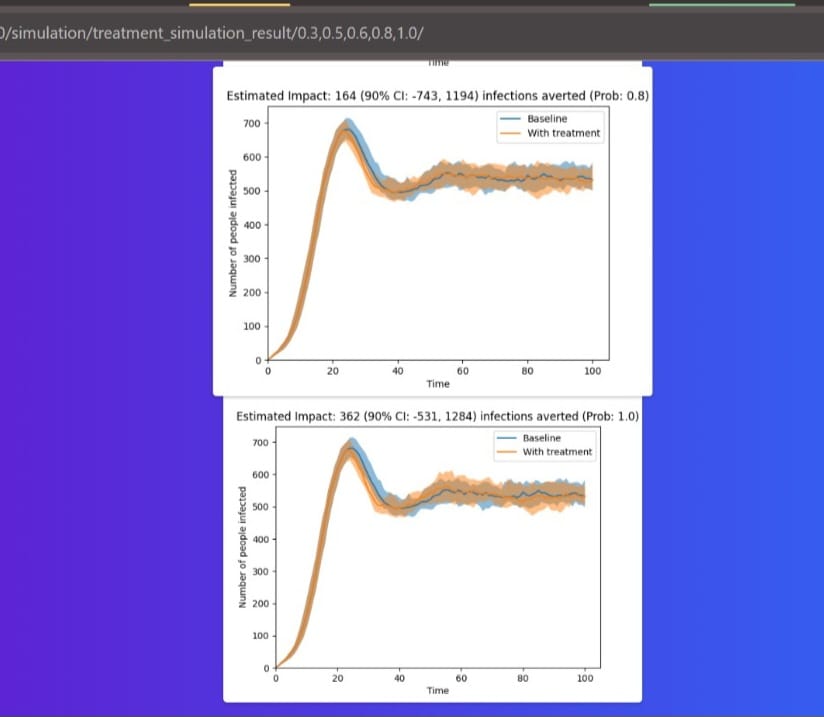
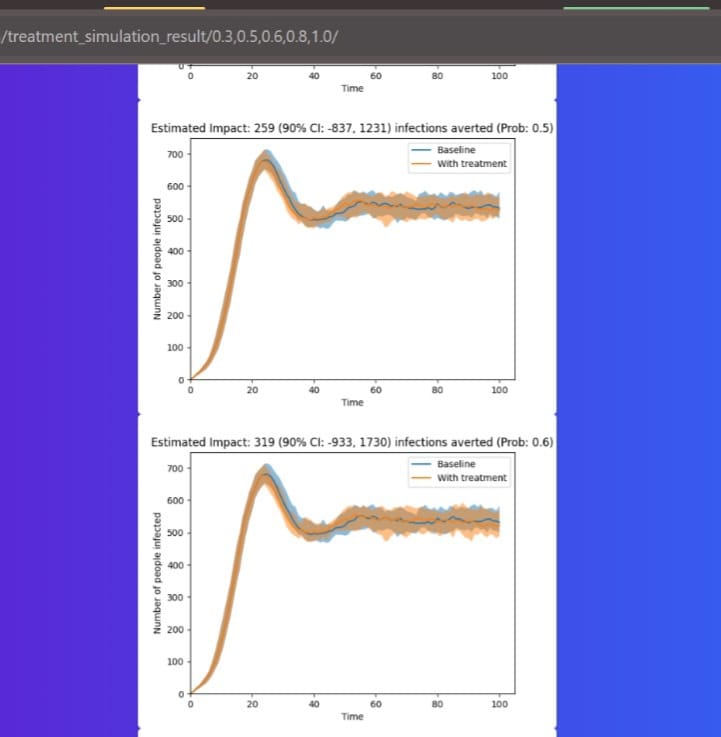
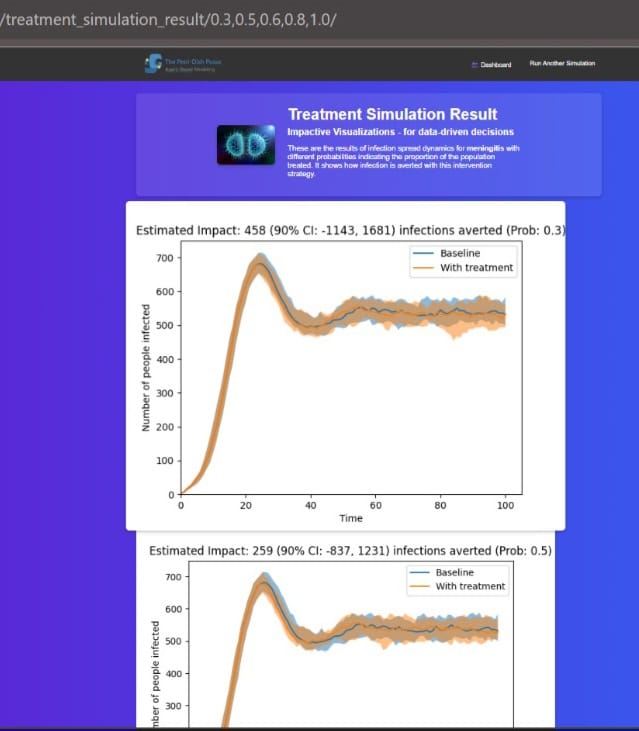


**Treatment Simulation**

Initializing different parameters



Visualization



## API Documentation

### Endpoints

* **/simulation/**: This endpoint accepts simulation parameters and starts the simulation process. It returns the results in a format suitable for visualization.

### Request and Response Formats

 **Request**:

* Data is sent as form data. The specific form and parameters depend on the simulation type selected.
* For instance, submitting a form for a normal simulation will require different parameters than submitting a form for a vaccination simulation.

 **Response**:

* The response includes the results of the simulation, such as time-series data for infection and recovery rates, and visualizations like graphs.

### Authentication

* Basic access does not require authentication.

### Error Codes

* **500 Internal Server Error**: Indicates an issue with the server or the simulation process.

## FAQ and Troubleshooting

### Common Questions

* **How do I start a simulation?**
  + Navigate to the appropriate form on the homepage, enter the required parameters, and submit the form.

### Troubleshooting Tips

* **Virtual Environment Issues**: Make sure the virtual environment is activated. If it's not, the application may fail to run correctly due to missing dependencies.
* **Dependency Problems**: If you encounter issues related to missing packages, re-run make install to ensure all required packages are installed.

## Contact and Support

### Support Channels

* **Email**: thepetridishposse@gmail.com
* **GitHub Issues**: Report issues and contribute to the project on https://github.com/janymuong/meningitis\_sim.git.

### Feedback

Please get in touch with Jany Muong at me.roumuong@gmail.com for feedback or suggestions.