Rubik’s Cube Documentation

This project is a Ruby on Rails application designed to simulate a 3x3x3 Rubik's Cube.

The core functionality is managed by a CubeRotationService class and API is implemented with CubeController.

The primary goal of this system is to provide a robust and testable implementation of Rubik's Cube mechanics.

Simulation is implemented using 6 3x3 matrix for each face of cube.

This implementation is more suitable for this task (implementing simulation only) then 3D matrix.

Implementation using 6 3x3 matrix is better for simulating Rubik’s Cube like this, also for solving algorithms.

Implementation using 3D matrix is better if cube needs to be visualized and animated.

Technologies

1. **Ruby:** Programming Language, 3.4,5 version
2. **Rails:** Web framework, Ruby on Rails, 8.0.2.1 version
3. **RSpec:** Framework for testing
4. **SQL Server:** Database for saving state of cube
5. **Postman:** Tool for testing API endpoints

Setup

1. Install technologies mentioned above
2. Ensure you have **Bundler** installed
   - *gem install bundler*
3. Clone repository
4. Install Dependencies
   - *bundle install* – this will look at *Gemfile* and install required gems/libraries
5. Setup Database
   - *rails db:migrate* – this will create database in SQL Server which is used by RubiksCube model
6. *Starting server – running API*
   - *rails s* – this will start rails server, API will be available at *http://localhost:3000*
7. Use Postman collection to test API endpoints and rotate cube
8. **Unit tests**

- *bundle exec rspec, or*
- *bundle exec rspec --format documentation*

API Endpoints

1. GET /api/v1/cube – Return state of cube in JSON

2. POST /api/v1/cube/rotate – Accepts *move* in body, and rotates cube based on selected move
   - Moves: R, R’, L, L’, B, B’, F, F’, U, U’, D, D’
   - R - Right, L – Left, F – Front, B – Back, U – Up, D – Down
   - “ *MOVE* ” – without ‘ is clockwise rotation
   - “ *MOVE’* ” – with ‘ is counterclockwise rotation

3. POST /api/v1/cube/reset – Reset cube to solved state