

(y/n) your product is effective to delivery the knowledge in computer science to the product users.

- Yes, my product represents graph coloring problem which is a well-known concept in the algorithms division of computer science.

user friendly/appealing in terms of the following criteria.

(y/n) The landing page is attractive. (hints: the homepages of the high-tech giants)

- Yes, trying to make the landing page very visually appealing like popular gaming platforms.

(y/n) Users are able to understand and play the puzzle game quickly.

- Yes, I provide instructions on how to play the game at the beginning of the game which will help the users to understand the game and play the game easily.

(y/n) Users can just jump in and start playing (trying out) the game immediately without the registration process.

- No, the user needs to register and login to play the game.

Your product should have the following functions.

(y/n) Users can register with a username and a password.

- Yes, users register so that they can login with username and password.

(y/n) The performance of registered users is updated after each trial and can be displayed upon requests.

- No, I don't want to implement this as this feature doesn't make sense and doesn't add any value in my product or it doesn't help the user in learning things. And more over this is not a multi-player game so the stats don't help the user.

(y/n) Users can ask for hints and/or solutions.

- Yes, I will provide hits to the user if he/she wants. Hits would like which color to select and which node to select.

(y/n) Administration account

Have all the functionality like the regular registered users.

- Yes, the admin ca login and do all regular user activities as well.

Have additional privileges likes user account removals or password-reset.

- Yes, admin can reset password for normal users, can add/remove users, block/unblock users.

(y/n) Do you have a brute-force method as the comparison basis for the puzzle solver.

- Yes, I have a brute force solution and even a greedy approach and backtracking.

(y/n) Do you have a better algorithm than brute-force.

- Yes, I have a greedy approach and backtracking.

Explain if the puzzle is targeted at a single user or multiplayer, competitive or non-competitive.

- It is a single player game, and it is a non-competitive game.

Explain how to deploy your product.

Prerequisites

- Have Git installed on your machine.
- Have a Heroku account. If you don't, sign up at Heroku's website.
- Install the Heroku CLI. Instructions can be found on Heroku's documentation.

Step 1: Prepare Your Flask Application

- Procfile: Create a file named Procfile in the root directory of your Flask application with the following content to tell Heroku how to run your app:
- web: gunicorn app:app
- Replace app:app with yourfilename:app if your Flask file is named differently.
- Requirements.txt: Ensure you have a requirements.txt file in your project root with all the necessary packages listed. You can generate one using: **pip freeze > requirements.txt**

Step 2: Create a Heroku Application

- Login to Heroku CLI: `heroku login`
- This command opens a web browser to log in or sign up.
- Create a Heroku App: `heroku create your-app-name`
- Replace your-app-name with a unique name for your app or leave it blank to let Heroku generate one for you.

Step 3: Add a Database

If your application uses a database:

- Add Heroku Postgres: `heroku addons:create heroku-postgresql:hobby-dev`
- This adds a free PostgreSQL database to your application.
- Set Database URI in Your App: Update your Flask app's configuration to use the `DATABASE_URL` environment variable provided by Heroku for your database connection.
- Set Database URI in Your App: Update your Flask app's configuration to use the `DATABASE_URL` environment variable provided by Heroku for your database connection.

Step 4: Deploy Your Application

Push to Heroku:

- `git push heroku main`
- If your main branch is named differently, replace `main` with your branch name.

Step 5: Ensure One Dyno is Running

- `heroku ps:scale web=1`
- This command ensures that at least one instance of your application is running.

Step 6: Open Your Application

- `heroku open`
- This command opens a web browser to your deployed application.

Additional Notes

- Environment Variables: Set any required environment variables using the Heroku dashboard or the CLI with `heroku config: set VAR_NAME=value`.
- Logs: Access logs for troubleshooting with `heroku logs --tail`.