Problem Statement: Flask Contact Form Application Challenge

Objective
Your task is to create a Flask web application that includes a contact form. This project will test your understanding of routing, blueprints, and WTForms with user input and form validation. By the end of this challenge, you will have a functional web application where users can submit contact messages.

Project Requirements
1. Project Directory Structure
Replicate the following directory structure for your project:

```
flask_app/
├── app/
│   ├── __init__.py
│   ├── main/
│   │   ├── __init__.py
│   │   ├── routes.py
│   │   ├── forms.py
│   │   └── templates/
│   │       ├── layout.html
│   │       ├── home.html
│   │       └── contact.html
│   ├── static/
│   │   ├── css/
│   │   └── js/
│   └── templates/
│       └── base.html
├── run.py
└── requirements.txt
```

2. Flask Application Setup
- Create a Flask application with a factory function in `app/__init__.py`.
- Configure a secret key for CSRF protection.

3. Blueprints
- Set up a blueprint for the main routes in `app/main/__init__.py`.
- Register the blueprint in the application factory.

4. Routes
- Create a route for the home page (`/`) in `app/main/routes.py`.
- Create a route for the contact form page (`/contact`) that handles both GET and POST requests.

5. WTForms
- Create a `ContactForm` class in `app/main/forms.py` with the following fields:
  - Name (StringField) - Required
  - Email (StringField) - Required and must be a valid email address
  - Message (TextAreaField) - Required
  - Submit (SubmitField)
- Validate the form and display appropriate flash messages upon successful submission.

6. Templates
- Create a base template (`app/templates/base.html`) that includes the basic HTML structure and blocks for title and content.
- Create a home page template (`app/main/templates/home.html`) that extends the base template.
- Create a contact form template (`app/main/templates/contact.html`) that extends the base template and includes the form.

7. Static Files
- Organize static files (CSS and JavaScript) in the `app/static/` directory as needed.

8. Running the Application
- Create a `run.py` file in the root directory to run the application.

9. Testing
- Run the application and ensure the following:
  - The home page displays correctly at `http://127.0.0.1:5000/`.
  - The contact form displays correctly at `http://127.0.0.1:5000/contact`.
  - The contact form validates input and displays flash messages upon submission.

Ensure your project follows the directory structure and requirements outlined above. Once completed, make sure you push your code to Git and initiate your PRs.

Tips
- Pay attention to file and directory structure.
- Ensure proper import statements and configurations.
- Test your application thoroughly to catch and fix any errors.

Good luck, and happy coding!