Project Title	📄 **Brain Tumor MRI Image Classification**
Skills Takeaway From This Project	• Deep Learning • Python • TensorFlow/Keras or PyTorch • Data Preprocessing • Transfer Learning • Model Evaluation • Streamlit Deployment
Domain	**Medical Imaging — Image Classification**

📌 Problem Statement

This project aims to develop a deep learning-based solution for classifying brain MRI images into multiple categories according to tumor type. It involves building a custom CNN model from scratch and enhancing performance through transfer learning using pretrained models. The project also includes deploying a user-friendly Streamlit web application to enable real-time tumor type predictions from uploaded MRI images.

📌 Real-Time Business Use Cases

1. **AI-Assisted Medical Diagnosis:**
   Provide radiologists with AI-powered tools to quickly classify brain tumors based on MRI images, reducing diagnostic turnaround time and increasing accuracy.

2. **Early Detection and Patient Triage:**
   Automatically flag high-risk MRI images for immediate specialist review,

improving hospital workflow and patient care prioritization.

3. **Research and Clinical Trials:**
   Use AI classification tools to segment patient datasets by tumor type, aiding in research studies and clinical trial recruitment.

4. **Second-Opinion AI Systems:**
   Deploy AI-powered classification tools in telemedicine or remote consultation setups for second-opinion diagnostics in under-resourced healthcare regions.

📌 **Project Workflow:**

1. **Understand the Dataset**

- Review the number of categories (tumor types) and sample images.
- Check for class imbalance and image resolution consistency.
- Explore image distributions visually.

---

2. **Data Preprocessing**

- Normalize pixel values to a 0–1 range.
- Resize images to a consistent shape suitable for model input (e.g. 224x224 pixels).

---

3. **Data Augmentation**

- Apply transformations like rotation, horizontal/vertical flipping, zoom, brightness adjustments, and shifts to artificially increase training data and improve model generalization.

---

4. **Model Building**

- **Custom CNN:** Design a convolutional neural network from scratch, selecting appropriate convolution, pooling, and dense layers.

- Implement dropout and batch normalization layers to avoid overfitting and stabilize learning.

---

5. **Transfer Learning**

- Load pretrained models (Example : ResNet50, MobileNetV2, InceptionV3, EfficientNetB0) with ImageNet weights.

- Replace the top classification layers with new dense layers suited for the tumor categories.

- Optionally unfreeze top layers for fine-tuning after initial training.

---

6. **Model Training**

- Train both custom CNN and transfer learning models.
- Use callbacks like EarlyStopping and ModelCheckpoint to monitor validation loss and save the best performing models.

- Track training and validation metrics.

---

7. **Model Evaluation**

- Evaluate models using metrics like accuracy, precision, recall, F1-score, and confusion matrix.
- Visualize model performance trends using training history plots for accuracy and loss.

---

8. **Model Comparison**

- Compare results of custom **CNN** vs **pretrained** models.
- Identify the most accurate, efficient, and reliable model for deployment.

---

9. **Streamlit Application Deployment**

- Build an interactive web application where users can upload brain MRI images.
- Display predicted tumor type along with model confidence scores.
- Ensure the UI is intuitive and informative.

📌 Dataset

- **Source:** Brain Tumor MRI Multi-Class Dataset

📌 Project Deliverables

1. Trained models: **custom CNN** and **pretrained models** (.h5).
2. Streamlit application for tumor classification.
3. Python scripts or notebooks for training, evaluation, and deployment.
4. Model comparison

5. Public GitHub repository with README.
6. Maintain clean, modular, and well-commented code.

🛠️ Technical Tags

`Deep Learning, Image Classification, Medical Imaging, Brain MRI Analysis, CNN, Transfer Learning, TensorFlow, Keras, PyTorch, Data Augmentation, Data Preprocessing, Model Evaluation, Performance Metrics, Streamlit Deployment, Confusion Matrix, Accuracy & Loss Visualization, Model Comparison, Healthcare AI, Computer Vision, Deployment Ready Applications, AI in Radiology`

⏳ Timeline

The project should be completed and submitted **within 14 days** from the date it is assigned.