Project Collaboration Guide: Git Workflow for Trika.ai

📂 Overview

This document explains the Git collaboration setup for the **Trika.ai** project. Our project has three contributors:

- **Kelvin** (You) ➔ Branch: kelvin-module
- **Varun** ➔ Branch: varun-module
- **Aditi** ➔ Branch: aditi-module

We follow a structured Git branching strategy to ensure clean collaboration and avoid conflicts.

⚖️ Workflow Summary

```
Main Branch (protected)
 └ Feature Branches
    ├ kelvin-module
    ├ varun-module
   └ aditi-module
```

Each developer works in their **own module branch**. When ready, they raise a **Pull Request (PR)** to merge into main, which is reviewed and approved before merging.

✍️ Steps to Follow (for Varun and Aditi)

✅ One-Time Setup:

1. **Clone the main repo:**

```
git clone git@github.com:<kelvin-username>/trika.ai.git

cd trika.ai
```

2. **Create your feature branch:**

```
# For Varun:

git checkout -b varun-module


# For Aditi:

git checkout -b aditi-module
```

3. **Push your branch to GitHub:**

```
git push -u origin varun-module   # or aditi-module
```

⚡ Regular Development Flow

1. **Start your work in your module branch:**

```
git checkout varun-module    # or aditi-module
```

2. **Pull latest changes from main regularly:**

```
git checkout main
git pull origin main

git checkout varun-module  # go back to your branch
git merge main             # merge updates
```

3. **Add and commit your changes:**

```
git add .
git commit -m "Added XYZ feature"
```

4. **Push to your branch:**

```
git push origin varun-module   # or aditi-module
```

5. **Create a Pull Request on GitHub:**

- Go to GitHub repo
- Click Compare & pull request
- Make sure base is main and compare is your-branch
- Add reviewer(s)
- Submit PR

6. **Review and Merge:**

- PR will be reviewed by team members
- Once approved, it will be merged into main

🔄 Visual Workflow Diagram

```
[ varun-module ]
      |
  git push origin
      |
   [ GitHub PR ] <--- Review <--- [ kelvin / aditi ]
      |
  Merge to [ main ]
```

Repeat the same for aditi-module and kelvin-module.

🏢 Folder Structure Recommendation

```
trika.ai/
 ├ backend/
 |   └ apps/
 |       ├ web/
 |       ├ mobile/
 |      └ wearable/
 ├ web/
 ├ mobile/
 └ wearable/
```

Each person focuses on a specific folder/module.

⚡ Pro Tips

- Do **not** push directly to main.
- Pull from main regularly to stay up-to-date.
- Always raise a PR to merge your work.
- Add meaningful commit messages.

Let’s collaborate efficiently and keep the codebase clean!