**Deployment & Management Guide for AzureCostApp Project**

**1. Project Structure on Local Laptop**

C:\Kractcostapplication\ # Root folder for the whole project

│

├─ frontend-react\azurecostapp-frontend-react\ # React frontend app folder

│

├─ api\azurecostapp-local\ # Backend API (Node.js) folder

│

└─ .github\workflows\ # GitHub Actions workflows folder

**2. Local Development**

**Running Frontend Locally**

cd frontend-react/azurecostapp-frontend-react

set PORT=8080 && npm start

* Runs the React frontend on localhost:8080

**Running Backend Locally**

cd api/azurecostapp-local

node server.js

* Runs the Node.js backend API locally

**3. Git Version Control Workflow**

**Add & Commit Changes**

git add <file-or-folder-path>

git commit -m "Your descriptive commit message"

Example:

git add frontend-react/azurecostapp-frontend-react/package.json

git commit -m "Change build script to disable CI errors"

**Push to Remote GitHub Repository**

git push origin main

*(Replace main with your branch name if different)*

**4. GitHub Actions Workflow for Azure Static Web Apps Deployment**

**Workflow File Location**

.github/workflows/azure-static-web-apps.yml

**Important Configurations**

app\_location: frontend-react/azurecostapp-frontend-react

api\_location: api/azurecostapp-local

output\_location: build

* app\_location: Path to your React app folder (relative to repo root)
* api\_location: Path to your API folder
* output\_location: Relative to app\_location where the build output is located

**Build Script in package.json (frontend)**

"scripts": {

"build": "CI=false react-scripts build",

"start": "react-scripts start",

...

}

* Using CI=false disables ESLint warnings as errors during build on Azure.

**5. Deploying Changes**

* Commit and push your code to GitHub main branch.
* GitHub Actions automatically triggers build & deploy.
* Check deployment logs in GitHub Actions tab.
* Access your app via the Azure Static Web Apps URL.

**6. Common Troubleshooting**

* **Build folder not found error**:  
  Make sure output\_location in workflow is relative to app\_location and is set to build.
* **ESLint errors stopping build**:  
  Use "build": "CI=false react-scripts build" in package.json to ignore warnings as errors.
* **Node version mismatch warnings**:  
  These are warnings; app can still build. Use .nvmrc or Azure config to specify Node version if needed.

**7. Useful Commands Summary**

| **Command** | **Description** |
| --- | --- |
| cd frontend-react/azurecostapp-frontend-react | Navigate to frontend app folder |
| set PORT=8080 && npm start | Run frontend locally on port 8080 |
| cd api/azurecostapp-local | Navigate to backend API folder |
| node server.js | Run backend locally |
| git add <file/folder> | Stage changes for commit |
| git commit -m "message" | Commit staged changes |
| git push origin main | Push commits to GitHub |
| npm run build | Build frontend locally |

**8. Additional Notes**

* Always commit your code regularly.
* Monitor GitHub Actions logs for build/deploy issues.
* The GitHub Actions workflow automates build and deployment for you on every push.

If you want, I can help you prepare this as a nicely formatted PDF or Markdown file for easy reference. Would you like that?