1. frontend-deployment.yaml

This file defines the Deployment for your frontend application. It manages your frontend Pods, sets up the environment variables to correctly point to the internal backend service, and defines resource limits.

Important: Just like the backend, ensure you have built your frontend Docker image and pushed it to a container registry before applying this.

```
# frontend-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend-deployment
  labels:
    app: frontend
spec:
  replicas: 1 # You can adjust this for more frontend instances
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
      - name: frontend
        # --- IMPORTANT ---
        # Replace 'your-docker-registry/your-frontend-image:latest' with the actual path
to your image
        image: your-docker-registry/your-frontend-image:latest
        ports:
        - containerPort: 3000 # The port your frontend application listens on
        env:
        # Since we want internal communication for development:
        # The frontend will now talk directly to the backend's internal Kubernetes Service.
        - name: REACT_APP_BACKEND_URL
          value: http://backend-service:8001 # Uses the internal backend-service name
and port
```

```
        - name: NODE_ENV
          value: production
        resources: # Resource limits, translated from your Docker Compose
          limits:
            memory: "512Mi" # Kubernetes uses Mi (mebibytes)
            # You can also add CPU requests/limits if desired, e.g.:
            # cpu: "500m" # 0.5 of a CPU core
```

2. frontend-service.yaml

This file defines the Service for your frontend. We use the NodePort type, which is perfect for development environments. It allows you to access your frontend from your local machine via the IP address of your Kubernetes node (e.g., Minikube IP) and a dynamically assigned port.

```
# frontend-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: frontend-service # The internal name for your frontend service
  labels:
    app: frontend
spec:
  selector:
    app: frontend # This links the Service to the Pods created by the
frontend-deployment
  ports:
    - protocol: TCP
      port: 80       # The port this Service will listen on (standard HTTP)
      targetPort: 3000 # The port on the frontend container to forward traffic to
  type: NodePort      # This type exposes the service on a static port on each node's IP.
                      # Kubernetes will assign a port in the 30000-32767 range.
                      # You'll find this port using 'kubectl get service frontend-service'.
```