Step-by-Step Deployment Guide  
Full-Stack Microservices Application on Minikube

Prepared by: Mohammad Hassaan Ejaz

Institution: FAST NUCES Islamabad

Date: October 2025

# Table of Contents

[Automatically generated in Word after applying heading styles]

# Assignment Requirements Checklist

✅ 1. Architecture Requirements  
- Frontend, Backend, Database, Authentication Services  
✅ 2. User Authentication Module  
- Signup, Login, Forgot Password  
✅ 3. Technology Stack  
- React, Node.js/Express, PostgreSQL, JWT + Redis  
✅ 4. Containerization (Docker)  
- Individual Dockerfiles, Docker Compose  
✅ 5. Kubernetes Orchestration  
- Deployment & Service YAMLs, 3 replicas, NodePort  
✅ 6. Documentation  
- Architecture Diagram, Deployment, Access Instructions

# Step 1: Prerequisites Installation

Install Docker Desktop, Minikube, and kubectl. Verify installations using terminal commands:  
  
docker --version  
minikube version  
kubectl version --client  
  
[Insert Screenshot: Docker and Minikube verification]

# Step 2: Project Setup

Clone or download the project and verify the directory structure.  
  
[Insert Screenshot: Project structure directory tree]

# Step 3: Docker Compose Deployment

Run ./deploy.sh docker to start multi-container setup.  
Verify running containers using docker-compose ps.  
  
[Insert Screenshot: docker ps output]

# Step 4: Kubernetes Deployment on Minikube

Start Minikube, build images, and apply YAML files. Verify pods and services.  
  
[Insert Screenshot: kubectl get pods]  
[Insert Screenshot: kubectl get svc]

# Step 5: Testing and Verification

Run ./test.sh to validate functionality.  
Perform manual UI and API testing for signup, login, and password reset.  
  
[Insert Screenshot: Application login page]

# Step 6: Architecture Verification

Verify microservices independence and scaling using kubectl get deployments.  
  
[Insert Screenshot: Deployment replica verification]

# Step 7: Monitoring and Management

Use kubectl logs and kubectl top commands for logs and resource monitoring.  
  
[Insert Screenshot: Resource usage dashboard]

# Step 8: Cleanup

Remove Kubernetes and Docker resources using ./deploy.sh cleanup commands.  
  
[Insert Screenshot: Cleanup confirmation]

# Step 9: Documentation Verification

Ensure architecture diagrams, deployment steps, and access instructions are complete.  
  
[Insert Screenshot: Architecture diagram overview]

# Step 10: Demo Video Script

Outline includes introduction, deployments, testing, architecture verification, and cleanup demonstration.

# Final Verification

Project meets all assignment requirements:  
1. Microservices Architecture (Frontend, Backend, Auth, Database)  
2. User Authentication (Signup, Login, Forgot Password)  
3. Technology Stack (React, Node.js, PostgreSQL, JWT)  
4. Containerization (Dockerfiles, Docker Compose)  
5. Kubernetes Orchestration (3 Replicas per service)  
6. Documentation (Architecture, Deployment, Testing)  
  
[Insert Screenshot: Final deployment status]