# Test Buddy - Phase 0 Report (Setup and Deployment)

This document summarizes all activities, installations, and configurations completed during Phase 0 of the Test Buddy project. Phase 0 established the foundation for a full-stack AI-powered testing platform using Next.js for the frontend, FastAPI for the backend, and integrated deployments on Vercel and Render.

## 1. Project Directory Structure

The project was initialized with a unified folder structure to support monorepo-style development:  
  
C:\Users\91991\Desktop\Test Buddy  
│  
├── apps  
│ ├── web → Next.js Frontend (Deployed on Vercel)  
│ └── server → FastAPI Backend (Deployed on Render)  
│  
├── package.json  
├── README.md  
└── requirements.txt (Backend dependencies)

## 2. Tools and Dependencies Installed

* \*\*Frontend (Next.js)\*\*
* • Installed Next.js (JavaScript, not TypeScript)
* • Tailwind CSS setup with PostCSS and autoprefixer
* • Components created: Header, ModuleCard
* • Environment variable added: NEXT\_PUBLIC\_API\_BASE
* \*\*Backend (FastAPI)\*\*
* • Created FastAPI app (main.py) with CORS enabled
* • Added routers: health, sessions, planner, designer, author, executor, curator
* • Installed dependencies using requirements.txt
* • Verified local run using Uvicorn
* \*\*System Installations\*\*
* • Node.js and npm installed
* • Python (3.11+) with virtual environment
* • Git initialized and connected to GitHub repository (Ranjithliftoff/test-buddy)

## 3. Deployment Setup

* \*\*Frontend Deployment (Vercel)\*\*
* • Linked GitHub repository to Vercel
* • Successfully deployed frontend at:
* https://test-buddy-4ozk.vercel.app
* \*\*Backend Deployment (Render)\*\*
* • Configured FastAPI service using the free Render plan
* • Build command: pip install -r requirements.txt
* • Start command: uvicorn main:app --host 0.0.0.0 --port 10000
* • Verified health endpoint:
* https://test-buddy-b110.onrender.com/health/

## 4. Integration Verification

• Added backend health badge in the frontend (page.js) to dynamically check API connectivity.  
• Confirmed successful CORS communication between Vercel (frontend) and Render (backend).  
• Verified live connection via health check: 🟢 Backend: OK.

## 5. Phase 0 Summary

Phase 0 established the complete environment and verified full-stack connectivity. The project is now ready for Phase 1, where functional modules such as SourceSelector, ResultPanel, and ReviewBar will be implemented to handle user interactions and AI-driven insights.