## **Aim**

To deploy a full-stack Task Manager application from GitHub to Vercel and implement automated CI/CD for continuous integration and deployment using GitHub Actions.

## **Theory**

### **1. Git and GitHub**

* **Git** is a distributed version control system that tracks changes in source code during software development.
* **GitHub** is a cloud-based platform built on Git that allows multiple developers to collaborate, store, and manage code repositories.
* Key operations:  
  + git add → stages changes.
  + git commit → saves snapshots of the code.
  + git push → uploads local commits to GitHub.
  + git pull → fetches updates from the remote repository.
* Benefits: Version history, collaboration, rollback capabilities, and integration with CI/CD pipelines.

### **2. Vercel Deployment**

* **Vercel** is a cloud platform for deploying web applications, including static websites, React, Next.js, and Node.js apps.
* Features:  
  + **Automatic Builds:** Vercel detects the project type and runs build commands automatically.
  + **Instant Deployments:** Provides a live URL immediately after deployment.
  + **Environment Variables:** Allows secure configuration for sensitive data like database URLs and API keys.
* Advantages:  
  + No manual server setup.
  + Scalable deployments.
  + Seamless integration with GitHub for continuous deployment.

### **3. Continuous Integration (CI)**

* CI is a development practice where developers **frequently merge code changes** into a central repository.
* Automated processes run **tests, linting, and builds** to ensure code quality and prevent integration issues.
* Tools: GitHub Actions, Jenkins, Travis CI, CircleCI.

### **4. Continuous Deployment (CD)**

* CD automates the **release of code to production** once it passes the CI pipeline.
* Eliminates manual deployment errors and ensures the live application is always up-to-date.
* Vercel acts as the CD platform in this experiment: every push to GitHub triggers a new build and deployment automatically.

### **5. GitHub Actions**

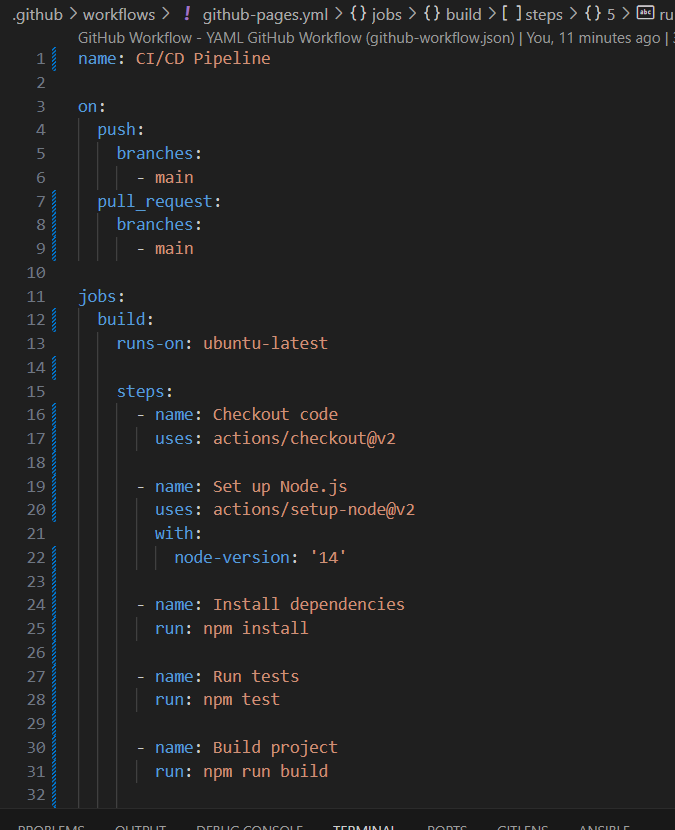
* GitHub Actions is a **workflow automation tool** integrated into GitHub repositories.
* Workflow is defined in .github/workflows/\*.yml files.
* Steps in a typical workflow:  
  1. Checkout code.
  2. Set up runtime (Node.js, Python, etc.).
  3. Install dependencies.
  4. Run tests.
  5. Build project.
  6. Deploy to cloud platform (like Vercel).
* Benefits: Fully automated CI/CD, integrated with GitHub, reduces manual intervention.

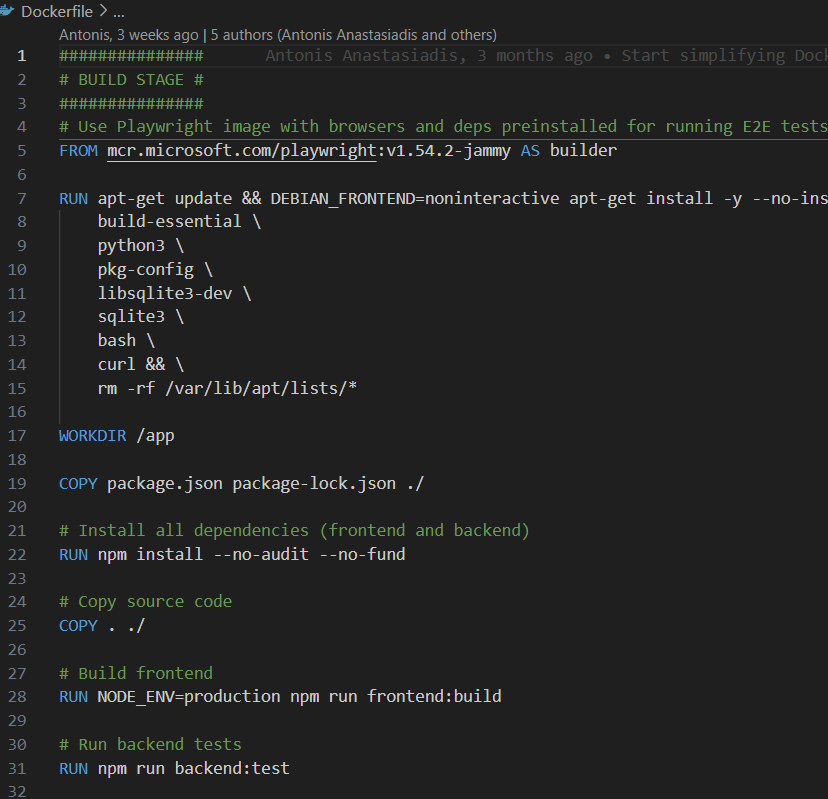
### **6. Full-stack Deployment Workflow**

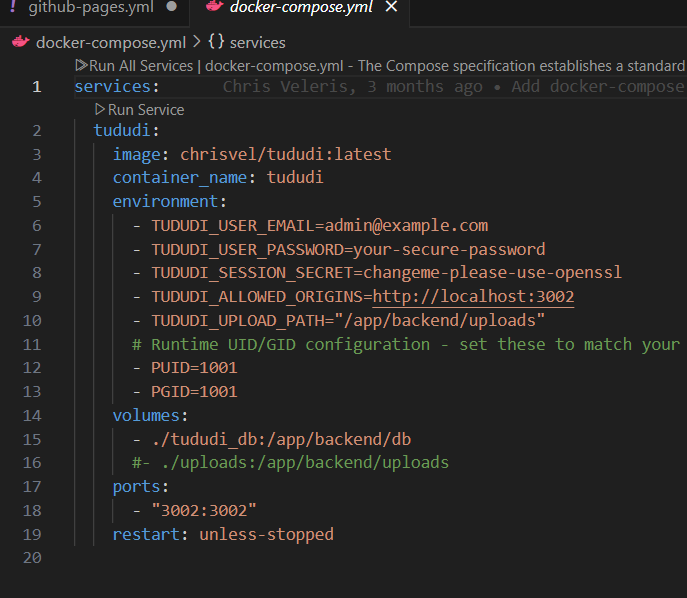
1. **Clone the repository** locally using git clone.
2. **Install dependencies** using package managers like npm.
3. **Push the project** to GitHub repository.
4. **Connect GitHub repo to Vercel** for deployment.
5. **Configure environment variables** (JWT secret, database URL, etc.).
6. **Automated CI/CD** triggers on code push:  
   * Builds the application.
   * Runs tests.
   * Deploys to Vercel automatically.
7. **Access the live application** via the Vercel-provided URL.

### **7. Advantages of this Setup**

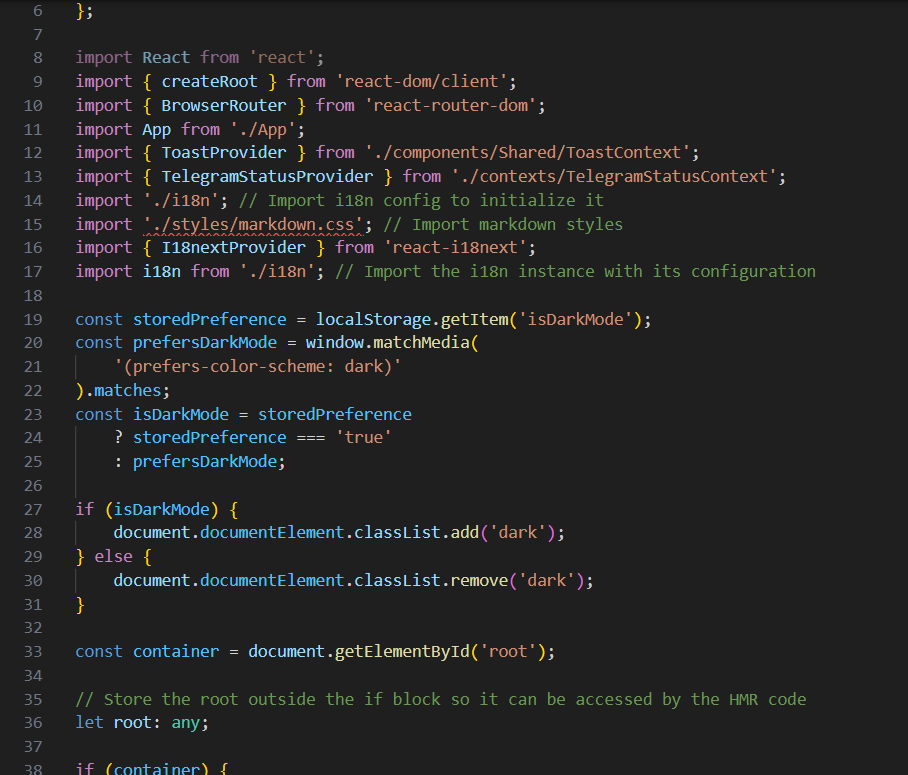
* Automation reduces human error.
* Live application always reflects the latest code.
* Easy collaboration and code review through GitHub.
* Rapid deployment and rollback capabilities.
* Modern software development practice aligned with industry standards.

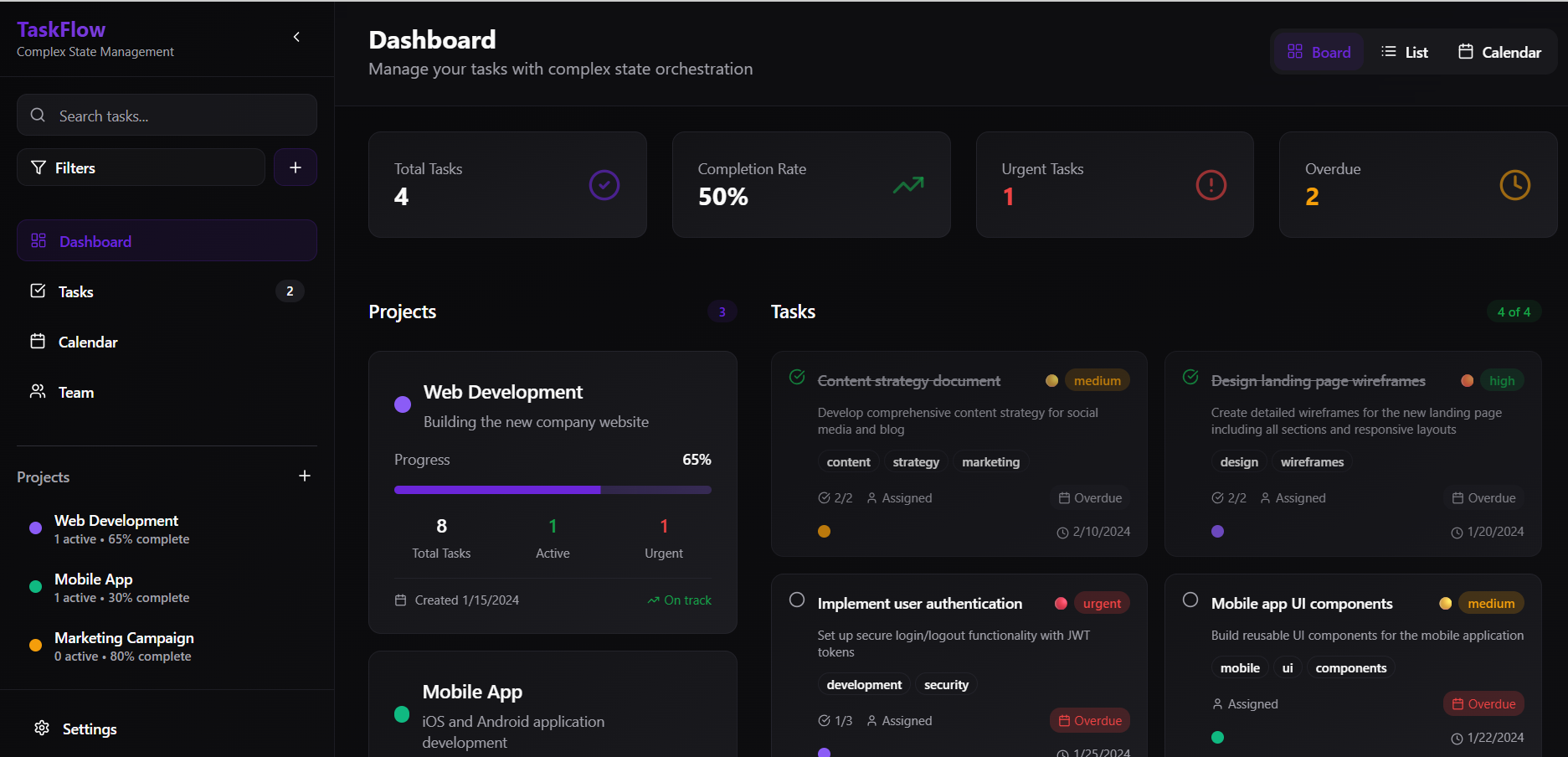
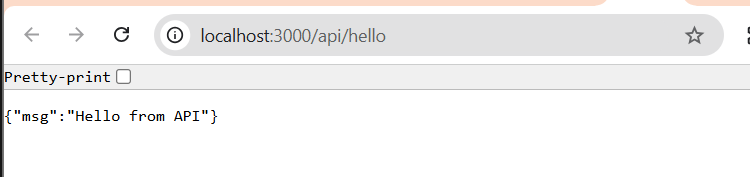
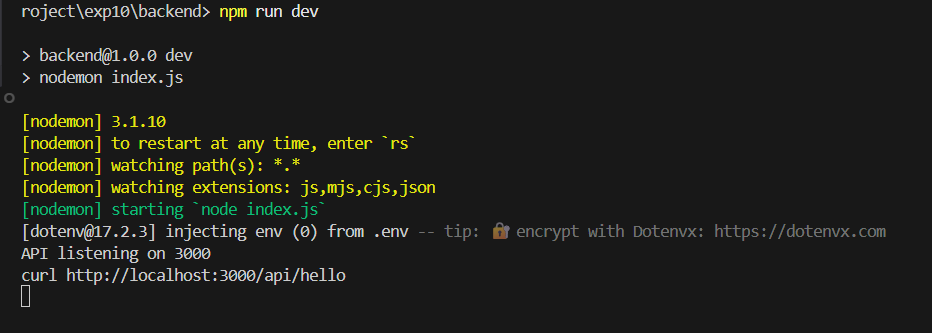
CI/CD WORKFLOW- .yml file

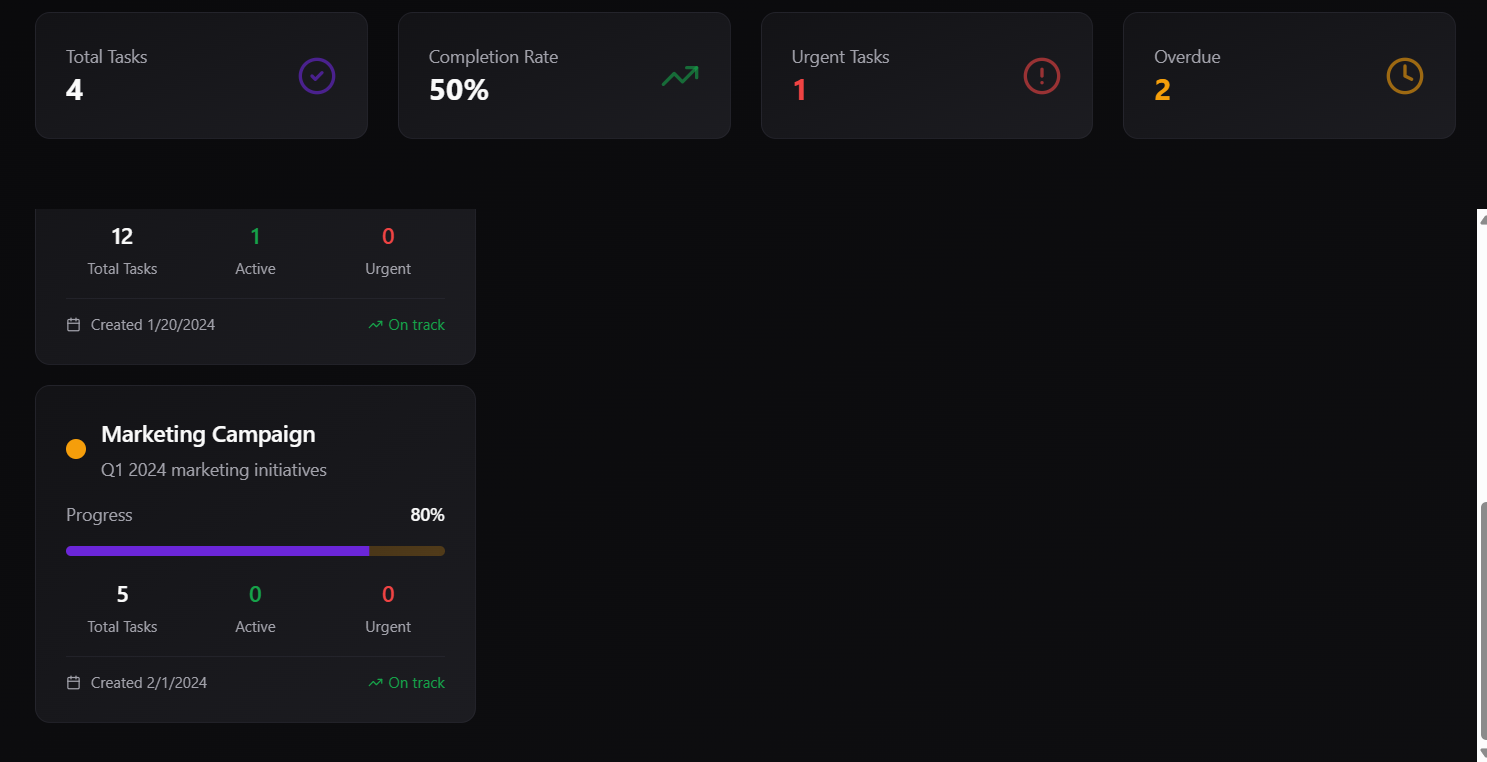
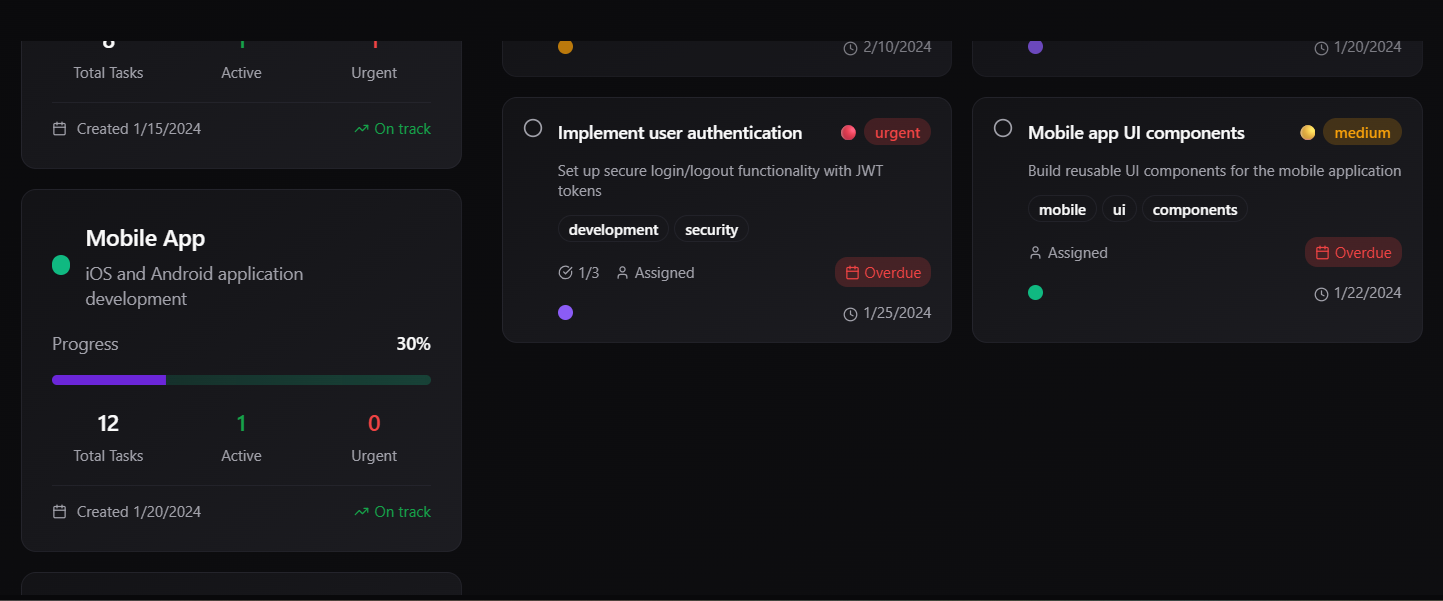
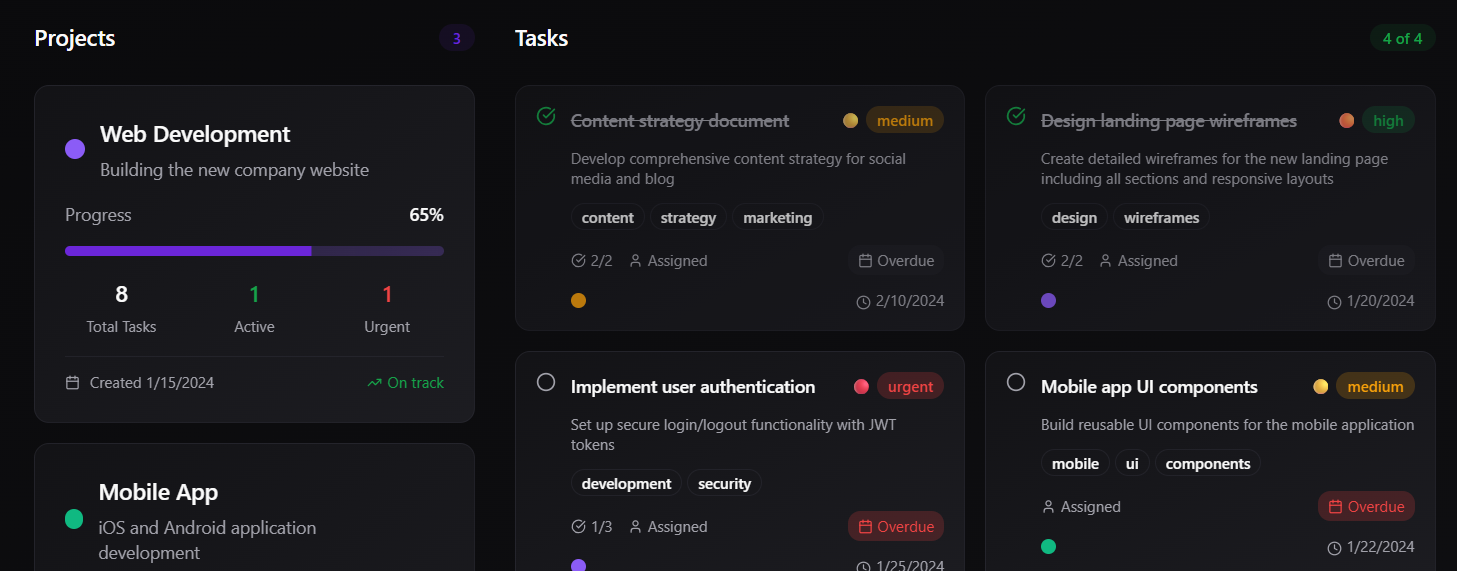
Dockerfile

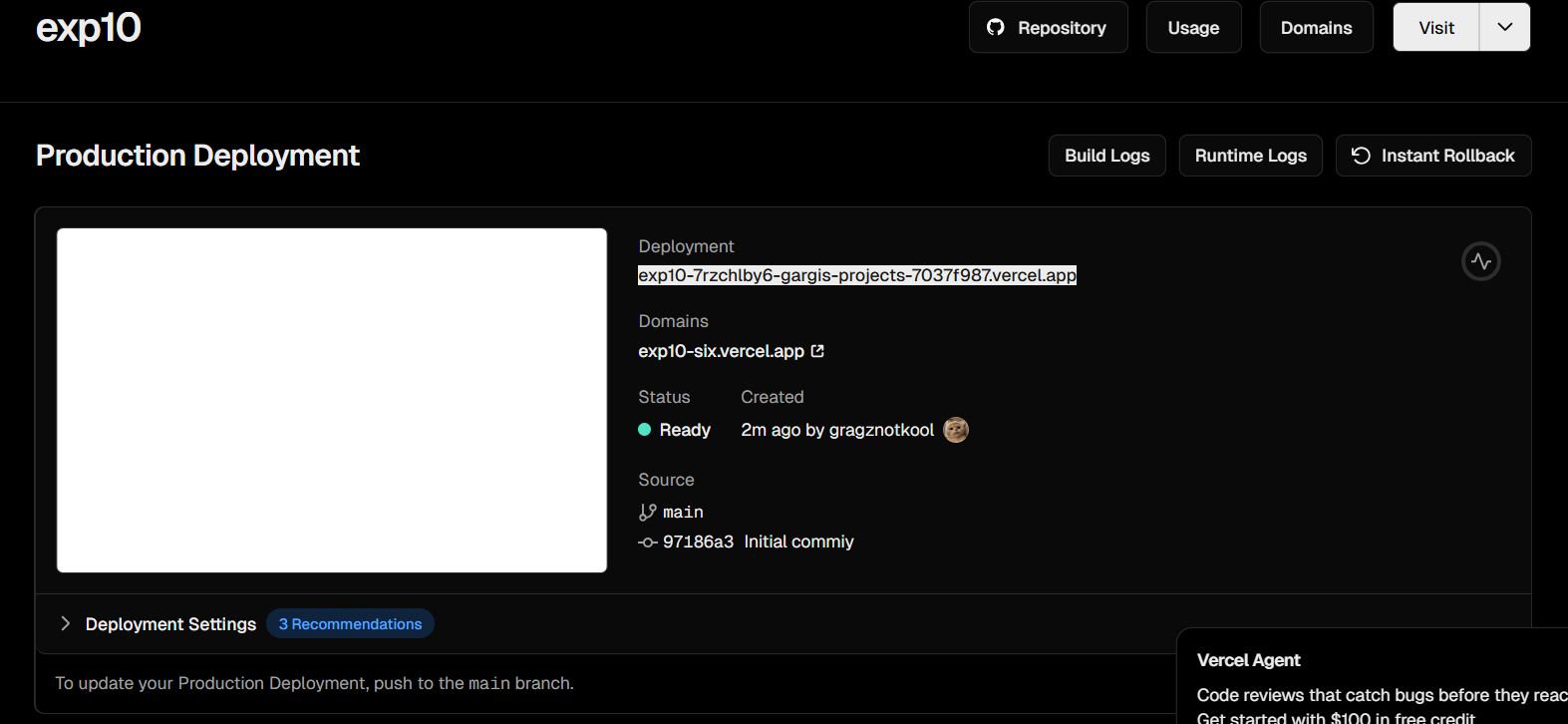
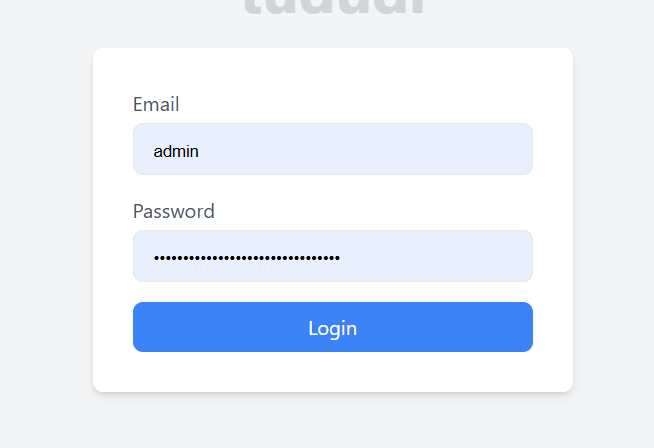
Docker-compose.yml 

Index.tsx



App.tsx





https://exp10-7rzchlby6-gargis-projects-7037f987.vercel.app/

## 

## **Conclusion**

* The Task Manager project was successfully deployed to Vercel from GitHub.
* Continuous deployment ensures that any changes pushed to GitHub automatically reflect on the live application.
* CI/CD integration improves development efficiency, reduces human error, and allows real-time updates of the application.
* This experiment demonstrates modern cloud deployment practices and the importance of version control, automated testing, and continuous delivery in software development.