Pomodoro Study Timer

Role: Designer & Developer
Timeline: March 2025
Type: Personal Project

Project Overview

Full-stack study timer application implementing the Pomodoro Technique with 25/5/15 minute work/break intervals. Built with React and deployed on Azure Static Web Apps, featuring a custom animated background and responsive design optimized for both mobile and desktop use.

Phase 1: Motivation & Research

Developed this project to gain hands-on experience with Figma design workflows and React development while creating a personal productivity tool.

Inspiration & Research

- **Primary Reference:** Analyzed pomofocus.io for core functionality and user experience patterns
- **Personal Use Case:** Drew from extensive personal experience using pomodoro timers to understand essential features
- **Design Inspiration:** FIFA 22 home page aesthetic for visual design language

Key Requirements Identified

- Clear state management with visual feedback (start button color changes, active state indicators)
- Proper interval transitions between work and break periods
- Responsive design for cross-device usage

Phase 2: Design & Planning

Design-first approach using Figma to establish visual foundation before development.

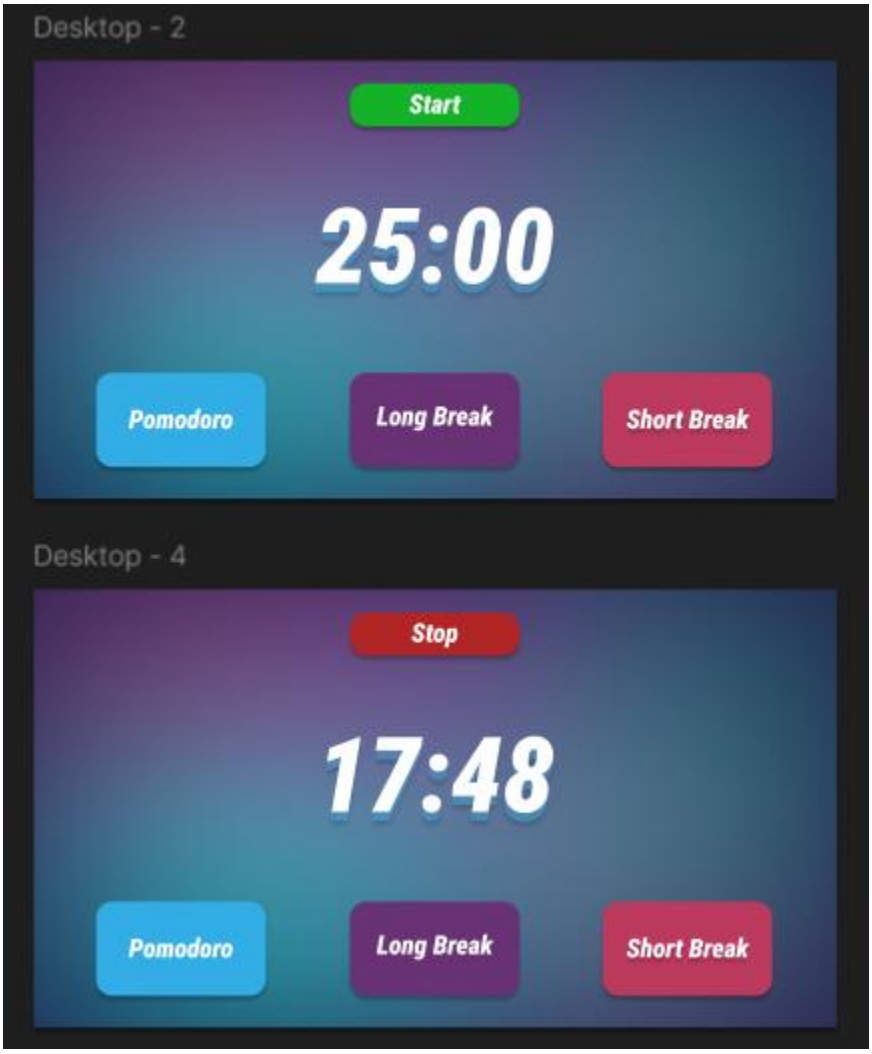
Design Process

- **Tool:** Figma for initial design and prototyping
- **Approach:** Single iteration design process with minimal revisions
- **Visual Style:** Inspired by FIFA 22 interface with custom animated blob background

- **Responsive Strategy:** CSS-based responsive design with mobile-first considerations

Core Features Defined

- Timer display with MM:SS format
- Three timer modes: Pomodoro (25min), Short Break (5min), Long Break (15min)
- Session progress tracking
- Visual state indicators for active/inactive states



Phase 3: Technical Implementation

Built using React with custom state management and deployed via Azure Static Web Apps.

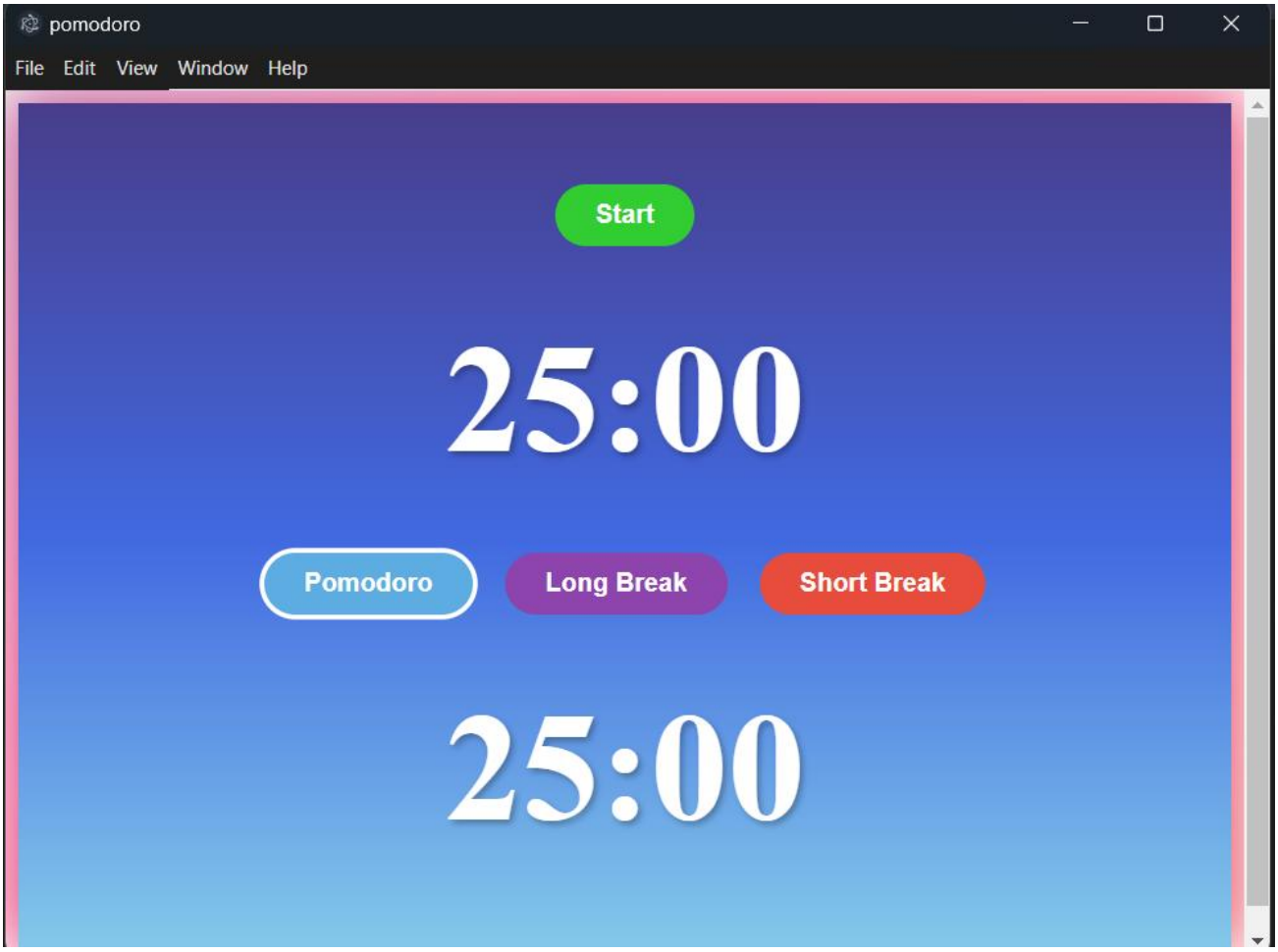
Technical Stack

- **Frontend:** React, HTML/CSS, JavaScript
- **Build Tool:** Webpack

- **Deployment:** Azure Static Web Apps
- **Animation:** Custom Canvas API implementation

Key Technical Challenges

- **State Management:** Complex timer logic requiring careful tracking of current state, remaining time, and transition logic
- **React Learning Curve:** First major React project, focusing on useState and useEffect hooks
- **Deployment Configuration:** Ensuring Azure Static Web Apps properly integrated with Webpack build process



Core Implementation Features

// State management for timer functionality

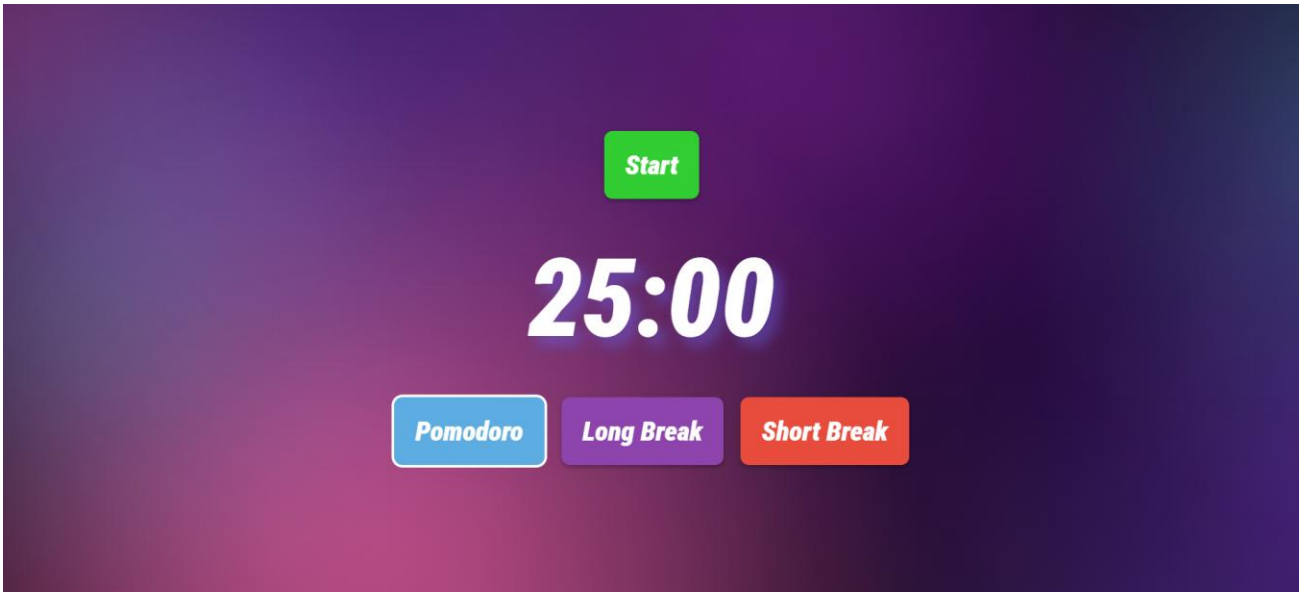
const [isActive, setIsActive] = useState(false);

const [timerType, setTimerType] = useState("pomodoro");

const [timeRemaining, setTimeRemaining] = useState(25 * 60);

const [cyclesCompleted, setCompleted] = useState(0);

- **Timer Logic:** useEffect hook managing interval updates and state transitions
- **UI State Synchronization:** Visual feedback system reflecting timer state changes
- **Custom Background:** Animated blob system using Canvas API with 25 floating elements



Phase 4: Testing & Refinement

Informal testing and community feedback leading to mobile layout improvements.

User Feedback Integration

- **Mobile Layout Issues:** Community feedback identified button alignment problems on mobile devices
- **Resolution:** Redesigned button positioning and spacing for improved mobile experience
- **Responsive Testing:** Verified functionality across multiple device sizes

Deployment Process

- **Platform:** Azure Static Web Apps (chosen for free tier and learning opportunity)
- **Challenge:** Webpack configuration integration with Azure build process
- **Solution:** Proper configuration of build pipeline for React application deployment

Technical Achievements

- **React Proficiency:** Gained hands-on experience with React hooks and state management
- **Modern Deployment:** Successfully deployed first application to cloud platform
- **Custom Animation:** Implemented complex Canvas-based background animation system
- **Responsive Design:** Created mobile-optimized interface using CSS media queries

Reflection

This project served as an excellent introduction to React development and cloud deployment. The biggest challenge was mastering React's state management for timer functionality, which required careful consideration of component lifecycle and state updates. The combination of design-first approach in Figma with React implementation provided valuable experience in translating designs to functional applications.

The project reinforced the importance of user testing, as community feedback revealed critical mobile usability issues that weren't apparent during desktop development. Successfully deploying to Azure Static Web Apps provided practical experience with modern development workflows and cloud platforms.