

# The Deadline Crusher

## Team Project

A productivity-focused web application  
designed to help students **crush deadlines early**

## **Team Members**

- **Landen Tomlin** – Team Leader
- **Josh Day** – Backend Development, Database Design
- **Tanner Andrews** – Frontend Development, Calendar & Time Tracking
- **Xander Murphy** – Frontend Development, UI Customization

# Project Description

The Deadline Crusher helps students manage assignments, deadlines, and daily responsibilities in a structured and motivating way.

The application combines:

- Smart task organization
- Calendar-based planning
- Gamified customization rewards

## Problem Domain

Many students struggle with:

- Managing multiple deadlines across courses
- Knowing which tasks are most urgent
- Planning work over time
- Staying motivated to complete tasks

Traditional to-do lists lack prioritization and engagement.

## Our Solution

The Deadline Crusher solves these problems by:

- Automatically organizing tasks into smart lists
- Visualizing deadlines using a calendar interface
- Rewarding productivity with unlockable customization options

## Core Features Overview

- Calendar / Time-Based Checklist
- Smart Lists & Custom Views
- Gamified Customization & Rewards

Built using a **3-tier architecture**

# **Feature 1: Calendar / Time-Based Checklist**

**Owner:** Tanner Andrews

## **Description**

A calendar interface that helps users plan and track tasks over time.

## **Requirements**

- Interactive calendar interface
- Tasks associated with due dates
- Upcoming and overdue indicators
- Time tracking per task

## **Feature 2: Smart Lists & Custom Views**

**Owner:** Josh Day

### **Description**

Automatically generated task lists and customizable views.

### **Requirements**

- Auto lists:
  - Today
  - Upcoming
  - Overdue

## Feature 2 Requirements Cont.

- Filters:
  - Tag
  - Priority
  - Due date
- Sorting options
- Search bar
- Save and pin custom views

## **Feature 3: Gamified Customization**

**Owner:** Xander Murphy

### **Description**

Users unlock visual customization rewards by completing tasks.

### **Requirements**

- Unlockable fonts, backgrounds, and themes
- Progress tracking tied to task completion
- Customization menu
- Persistent unlocked rewards

# Non-Functional Requirements

- Three-tier architecture
- Web-based application
- Responsive UI (desktop & mobile)
- Persistent user data
- Scalable and maintainable codebase
- Intuitive and consistent interface

# Total Features and Requirements

Features: 3

Requirements: 14

# Schedule & Milestones

## Sprint 1

- Project setup (React, Node.js, Express)
- Core task CRUD functionality
- Smart lists (Today / Upcoming / Overdue)
- Initial calendar UI
- Database schema design

# Schedule & Milestones

## Sprint 2

- REST API expansion
- Time tracking functionality
- Filters, sorting, and saved views
- Customization unlock system
- UI polish and testing

## Team Rules

1. Communicate effectively and consistently.
2. All weekly updates must be finished by Saturday of the current week.
3. Communication will be handled over discord weekly for quick messages and questions. Full meetings have been discussed, but days and times have not been established.

**Questions?**