



COLLEGE CODE : 9628

**COLLEGE NAME: UNIVERSITY COLLEGE OF ENGINEERING
NAGERCOIL.**

DEPARTMENT :COMPUTER SCIENCE AND ENGINEERING

STUDENT-ID : 3791FE0EE3ABCB99BA4438D4CDA7A1E8

REGISTER NO: 962823104078

DATE : 29-09-2025

COMPLETED THE PROJECT NAMED AS :

THEME CUSTOMIZER IN WORDPRESS

SUBMITTED BY ,

NAME : MOHAMMED RASIDH M

MOBILE NO : 6381482861

MVP IMPLEMENTATION

TITLE : THEME CUSTOMIZER IN WORDPRESS

1. Project Setup

Plugin Structure

Create main plugin directory: theme-customizer/

Set up folder structure:

```
theme-customizer/  
├── assets/  
|   ├── css/  
|   ├── js/  
|   └── images/  
├── includes/  
|   ├── admin/  
|   ├── frontend/  
|   └── customizer/  
├── templates/  
└── theme-customizer.php (main plugin file)
```

Development Environment

- Install WordPress locally (XAMPP, Local, or Docker)
- Set up PHP debugging tools (Xdebug)
- Install browser DevTools for frontend testing
- Configure text editor/IDE (VS Code with WordPress extensions)

Plugin Header Setup

- Define plugin metadata (name, version, author, description)
- Add WordPress hooks and filters
- Register plugin activation/deactivation functions
- Set up autoloading for classes

2. Core Features Implementation

Customizer Panel Registration

- **Color Customization**
 - Primary color picker
 - Secondary color picker
 - Background color options
 - Text color controls
 - Link color settings
- **Typography Controls**
 - Font family selection (Google Fonts integration)
 - Font size sliders for headings (H1-H6)
 - Font size for body text
 - Font weight options
 - Line height controls
 - Letter spacing adjustments
- **Layout Settings**
 - Site width options (boxed/full-width)

- Sidebar position (left/right/none)
- Header layout variations
- Footer layout options
- Content padding controls

- **Header Customization**

- Logo upload
- Site title/tagline toggle
- Header height control
- Navigation menu styling
- Header background options

- **Footer Customization**

- Footer widget areas
- Copyright text editor
- Social media links
- Footer background styling

Live Preview Functionality

- Implement real-time preview using WordPress Customizer API
- Add postMessage transport for instant updates
- Create selective refresh for complex elements
- Implement preview iframe communication

Settings API Integration

- Register all settings with WordPress Settings API
- Add sanitization callbacks for each setting
- Implement validation functions
- Set default values for all options

3. Data Storage (Local State / Database)

WordPress Options Table

- Store customizer settings in wp_options table
- Create unique option names with prefix (e.g., tc_primary_color)
- Use serialized arrays for grouped settings
- Implement options caching for performance

Custom Database Tables (if needed)

- Create custom table for advanced configurations:

```
CREATE TABLE wp_theme_customizer_presets (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  preset_name VARCHAR(255),  
  settings LONGTEXT,  
  created_at DATETIME,  
  updated_at DATETIME  
)
```

Data Structure

- **Settings Organization:**
 - Colors: Array of color values
 - Typography: Nested arrays for each element
 - Layout: Dimension and positioning values
 - Media: Image URLs and IDs

Export/Import Functionality

- Create JSON export of all settings
- Import validation and sanitization
- Backup current settings before import
- Error handling for corrupted imports

4. Testing Core Features

Functional Testing

- **Color Changes:**
 - Verify color picker functionality
 - Test color application to elements
 - Check color contrast for accessibility
 - Validate hex/RGB/RGBA values
- **Typography:**
 - Test font loading from Google Fonts
 - Verify font size changes across devices
 - Check font weight rendering
 - Test fallback fonts
- **Layout:**
 - Test responsive behavior at different breakpoints
 - Verify sidebar positioning
 - Check header/footer layouts
 - Test content area width changes

Compatibility Testing

- Test with default WordPress themes (Twenty Twenty-Four, etc.)
- Test with popular third-party themes
- Verify compatibility with common plugins
- Check for JavaScript conflicts

Browser Testing

- Chrome/Edge (latest versions)
- Firefox (latest version)
- Safari (macOS and iOS)
- Mobile browsers (Chrome, Safari)

Performance Testing

- Measure page load times
- Check database query efficiency
- Monitor memory usage
- Test with GTmetrix or Google PageSpeed

Security Testing

- Validate all user inputs
- Test for XSS vulnerabilities
- Check nonce verification
- Test capability checks for admin functions
- Sanitize all output data

5. Version Control(GitHub)

Repository Setup

- Initialize Git repository
- Create .gitignore file:

node_modules/

.DS_Store

*.log

wp-config.php

.env

- Create README.md with project description
- Add LICENSE file (GPL v2 or later for WordPress)

Branch Strategy

- **main/master:** Production-ready code
- **develop:** Development branch
- **feature/[feature-name]:** Individual features
- **bugfix/[bug-name]:** Bug fixes

Commit Guidelines

- Use descriptive commit messages
- Follow conventional commits format:
 - feat: New features
 - fix: Bug fixes
 - docs: Documentation changes
 - style: Code formatting

- refactor: Code restructuring
- test: Testing additions

Documentation

- Create CHANGELOG.md
- Document installation instructions
- Add code comments for complex functions
- Create inline documentation (PHPDoc)
- Document all filters and hooks

Milestones & Issues

- Create GitHub Issues for each task
- Use labels (bug, enhancement, documentation)
- Set up project board for task tracking
- Create milestones for Phase 3 completion