**Natural Hazard Alert Application**

**(Implementation plan)**

### 

**Project Manager =>**

Mr. Fahad Maqbool

**Project Team =>**

Haris Ahmad (BSSE51F20R009) (Group leader)

Bakhtawar (BSSE51F20R004) (Group member)

Aqsa Waseem (BSSE51F20R034) (Group member)

| Week 1 | System Architecture Implementation  Set up the development environment and tools.  Develop the core system architecture based on the design.  Create a basic database schema.  Begin coding the backend infrastructure. |
| --- | --- |
| Week 2 | System Architecture Implementation  Continue refining the core system architecture.  Implement data retrieval and storage mechanisms.  Establish communication between the frontend and backend.  Ensure the system is ready for data integration. |
| Week 3 | User Interface Implementation  Start developing the user interface (UI) for the mobile application.  Design the login and registration screens.  Begin working on the hazard alert display screen.  Ensure the UI is responsive on different devices. |
| Week 4 | User Interface Implementation  Continue UI development.  Design screens for user settings.  Implement navigation between different sections of the app.  Ensure consistent UI design across iOS and Android platforms. |
| Week 5 | Alert Generation Module  Begin implementing the alert generation algorithm for earthquakes.  Integrate a real-time earthquake data source.  Develop the UI components to display earthquake alerts.  Test the earthquake alert module for basic functionality. |
| Week 6 | Alert Generation Module  Expand the alert generation module to include floods.  Integrate real-time flood data sources.  Implement UI components for displaying flood alerts.  Test and validate the flood alert module. |
| Week 7 | Alert Generation Module  Add storm alert generation to the module.  Integrate real-time storm data feeds.  Develop UI components for storm alerts.  Test and validate the storm alert module. |
| Week 8 | User Account Management  Implement user account creation functionality.  Develop the user login and authentication system.  Ensure secure password handling and user data storage.  Begin work on user profile management features. |
| Week 9 | User Account Management  Continue refining user account management.  Develop features for updating user profiles.  Implement password reset and recovery options.  Test the user account management system for security. |
| Week 10 | Geographic Customization  Begin implementing geolocation functionality.  Integrate the mapping service for hazard visualization.  Develop the UI components for setting user location preferences.  Test geolocation accuracy and customization features. |
| Week 11 | Geographic Customization  Continue working on geolocation features.  Ensure that alerts are customized based on user location.  Test the mapping service for displaying hazard information visually.  Address any issues with geolocation accuracy. |
| Week 12 | Disaster Preparedness Information  Implement the section of the app that provides disaster preparedness information.  Populate the app with educational resources on safety measures.  Ensure easy navigation and accessibility of preparedness content.  Test the usability of the disaster preparedness section. |
| Week 13 | Testing and Quality Assurance  Conduct initial testing of the entire application.  Identify and document any bugs or glitches.  Begin addressing usability issues.  Share the app with a limited group of beta testers for feedback. |
| Week 14 | Testing and Quality Assurance  Address issues discovered during testing.  Conduct comprehensive testing on a wider scale.  Ensure cross-platform compatibility (iOS and Android).  Focus on optimizing performance and responsiveness. |
| Week 15 | Deployment and Post-Launch  Prepare the application for launch on the App Store (iOS) and Google Play Store (Android).  Perform final testing and quality checks.  Publish the application to both platforms.  Monitor user feedback and address any issues post-launch. |
| Week 16 | Post-Launch and Ongoing Maintenance  Continue monitoring user feedback and addressing issues.  Plan for ongoing maintenance and updates.  Document the project's implementation phase for reference.  Prepare for the transition to the maintenance phase. |