Portfolio Pro

Test Strategy and Plan for Agile

Table of Contents

1.0 Purpose of This Document 4

2.0 Introduction 4

2.1 Program/ Project Overview 5

2.2 Application Overview 5

3.0 Testing Objectives 6

4.0 Testing Scope 7

4.1 Scope of testing 7

4.2 In Scope 9

4.2.1 Program/Application / Module 9

4.2.2 Type of Testing 9

4.2.3 Other In-Scope Items 10

4.3 Out of scope 10

4.3.1 Program/Application / Module 10

4.3.2 Type of Testing 10

4.3.3 Other Out-of-Scope Items 11

5.0 Testing Approach & Strategy 11

5.1 Feature Testing Approach & Coverage 11

5.2 Regression Approach & Coverage 12

5.3 Integration Testing Approach & Coverage 12

5.4 Performance & Security Testing Approach & Coverage 13

5.5 Test Automation Approach & Coverage 13

5.6 Test Data Management 13

5.7 Testing Phases, Scope, Deliverables, Acceptance Criteria, Impacted Applications 14

6.0 Overall Testing approach 15

6.1 Test Design Approach 15

6.1.1 Design Methodology 15

6.1.2 Review Process 16

6.1.3 Entry Criteria for Test Design 16

7.0 Test Execution Approach 16

7.1 Execution Methodology 17

7.2 Test Data Management 17

7.3 Execution Cycles 18

8.0 Tools & Hardware / Software Requirements 18

8.1 Tools 19

8.2 Test Management Tool 19

9.0 Assumption 19

10.0 Dependencies 20

11.0 Defect Management 21

11.1 Defect Process Steps 21

12.0 Defect Severity & Priority Definition or Classification Guidelines 22

1. Purpose of This Document

The primary purpose of this document is to establish a comprehensive and actionable Quality Assurance (QA) testing strategy and plan specifically tailored for the **PortfolioPro web application**. With a precise and focused scope on the **Portfolio Creation (FRD Section 5.0)** and **Portfolio Page (FRD Section 6.0) modules**, this document serves as a foundational guide for the testing team. It meticulously outlines the systematic approach to quality assurance within an Agile development framework, ensuring that all defined functional and non-functional requirements within the in-scope areas are rigorously validated. This plan clearly defines:

- **Testing Objectives:** A clear articulation of the specific quality goals and aims to be achieved through the testing activities for the Portfolio Creation and Portfolio Page functionalities.
- **Testing Scope:** A precise delineation of the application areas that will undergo testing (in-scope: Portfolio Creation and Portfolio Page, along with their detailed sub-components), and an explicit identification of what lies outside this testing phase (out-of-scope).
- **Testing Approach and Strategy:** The methodologies, various types of testing (e.g., feature, regression, integration specific to these modules), and the overall strategic framework designed to ensure the highest quality delivery.
- **Roles and Responsibilities:** Though not detailed in this excerpt, a complete document would outline the specific roles and duties of team members involved in the testing process.
- **Resources:** Identification of the essential tools, testing environments, and human resources required to execute the defined testing activities efficiently and effectively.
- **Deliverables:** The tangible outputs and artifacts that will be produced as a result of the testing efforts, such as executed test case reports, comprehensive defect logs, and final test summary reports.
- **Entry and Exit Criteria:** The necessary conditions that must be rigorously satisfied to commence and conclude specific testing phases, ensuring a structured and controlled testing lifecycle.

2. Introduction

This Test Plan comprehensively outlines the QA testing strategy adopted to ensure the high-quality delivery of the **PortfolioPro web application**. The overarching objective is to rigorously validate that all features and crucial business rules pertinent to portfolio creation and management are implemented correctly and meet the meticulously defined functional requirements prior to their release. This document meticulously details the specific testing practices and procedures to be followed, encompassing the precise scope of testing, the tools to be utilized, the resources allocated, and the indicative timelines within the Agile development sprints.

The **PortfolioPro platform** is designed to empower users to effortlessly create, customize, manage, and share professional digital portfolios. The application's core aim is to deliver a seamless, intuitive, and interactive experience, fundamentally enhancing how individuals showcase their skills and accomplishments. It places significant emphasis on providing responsive design, offering a variety of themes, enabling dynamic content customization, and facilitating robust portfolio management, ultimately simplifying the process of professional self-presentation.

2.1 Program/ Project Overview

The **PortfolioPro project** is envisioned as an innovative web-based platform dedicated to streamlining the process of creating and showcasing digital portfolios. It is designed to be a comprehensive solution for individuals across various professions to present their work, experience, and skills in a professional and engaging manner. The program is structured to support a single primary user role – the portfolio creator – who possesses distinct functionalities centered around content generation, design selection, and sharing.

Estimated Benefits and Outcomes:

- **Enhanced Professional Presentation:** Provides users with a sophisticated and customizable online presence to showcase their expertise.
- **Streamlined Content Creation:** Simplifies the process of inputting personal, contact, education, work experience, and project details.
- **Diverse Design Options:** Offers multiple themes (Creative, Minimal, Modern) to cater to varied aesthetic preferences, enhancing user appeal.
- **Improved Shareability:** Facilitates easy sharing of portfolios via direct links, increasing visibility for users.
- **Intuitive User Experience:** Designed for ease of use, enabling even non-technical users to build professional portfolios independently.
- **Scalable Architecture:** Built with a foundation that allows for future expansion of features and user base.

Timelines: The PortfolioPro project is committed to an Agile development cycle, which inherently involves iterative development and continuous testing phases. Testing activities for the Portfolio Creation and Portfolio Page modules will be tightly integrated within each sprint, ensuring early feedback and rapid defect resolution. Specific sprint timelines will dictate the duration of each testing iteration.

2.2 Application Overview

PortfolioPro is a dynamic web-based application primarily focused on enabling robust digital portfolio creation and management. While the complete application ecosystem involves broader functionalities, this testing phase is concentrated on the following core modules and their integral sub-sections:

- **Portfolio Creation :** This is the multi-stage wizard where users systematically build their portfolio:
  - **Basic Information Page:** Captures foundational personal details (e.g., Full Name, Short Description, Job Title, Bio Title).

    - **Contact Page:** Gathers essential contact information (e.g., Email, Phone, Location, Social Links).
    - **Content Page:** Allows users to detail their academic achievements (Education), professional history (Work Experience), and showcased work (Projects).
    - **Image Page:** Facilitates the upload of visual assets, including Profile Photo and relevant Project Images.
    - **Theme Page:** Enables users to browse, preview, and select from predefined portfolio themes (Creative, Minimal, Modern) to personalize their portfolio's aesthetic.
- **Portfolio Page :** This is the final output and management hub for created portfolios:
    - Displays the fully rendered portfolio with all entered details and the selected theme.
    - Includes functionalities such as "Edit" (to revisit and modify portfolio content), "Copy Link" (to generate and copy a shareable URL), and "Share" (to disseminate the portfolio via various channels).

The application is meticulously designed to be highly responsive, ensuring optimal accessibility and usability across a diverse range of devices (desktops, tablets, and mobile phones), with a strong emphasis on intuitive navigation, data integrity, and visual appeal.

3. Testing Objectives

The primary objectives for the testing of the Portfolio Creation and Portfolio Page modules within the PortfolioPro application are meticulously defined to ensure a high standard of quality and user satisfaction. These objectives include:

- **Functional Compliance:** To comprehensively ensure that all features and business rules related to creating, editing, and viewing portfolios are implemented correctly and function precisely according to specified requirements. This encompasses validating the successful saving and retrieval of all user-entered data.
- **User Interface (UI) Integrity:** To confirm that the user interface is visually consistent, intuitive, and responsive across all portfolio creation steps and the final Portfolio Page. This involves verifying the presence, accurate rendering, and interactive functionality of all buttons, input fields, images, and theme-specific elements.
- **Input Data Validation :** To rigorously validate all input fields within the portfolio creation wizard (e.g., Basic Information, Contact, Content) for proper data type acceptance, format adherence (e.g., email format), and accurate error handling for invalid entries.
- **Input Size Limit Enforcement:** To verify that all defined character limits for text fields (e.g., Short Description, Project Name) and file size/type restrictions for image uploads (Profile Photo, Project Images) are correctly enforced, preventing data truncation or system errors.
- **Data Auto-Population Accuracy:** To confirm that any specified auto-population of user data (e.g., Full Name, Email) into subsequent fields or the final portfolio display functions correctly and accurately.
- **Navigation Flow Validation:** To ensure seamless and logical navigation between consecutive and previous steps within the portfolio creation process, as well as

smooth transitions to and from the Portfolio Page, without any data loss or redirection issues.

- **Theme Selection and Application Fidelity:** To verify that users can successfully select, preview, and apply the "Creative," "Minimal," and "Modern" themes, and that the chosen theme is accurately reflected and rendered on the final Portfolio Page.
- **Key Action Button Functionality:** To validate the precise functionality of all critical action buttons, including "Create Portfolio," "Edit," "Copy Link," and "Share," ensuring they perform their intended actions correctly and reliably.
- **Minimal Portfolio Creation Capability:** To confirm that the application allows for the successful creation and display of a portfolio using only the mandatory required fields, demonstrating system robustness for basic use cases.
- **Content Display Accuracy:** To verify that all sections of the portfolio (Basic Information, Contact, Content, Image) that a user has filled are correctly displayed and visible on the final Portfolio Page, including any added social media links.
- **Defect Identification and Management:** To systematically identify, document, categorize (severity and priority), track, and manage all software defects discovered within the in-scope modules, ensuring their timely resolution and retesting until closure.
- **Support for Development:** To provide constructive and timely feedback to the development team, actively contributing to the overall improvement of the PortfolioPro application's quality and stability.

4. Testing Scope

4.1 Scope of testing

The scope of testing for the PortfolioPro application in this phase is precisely defined to ensure comprehensive quality assurance for the **Portfolio Creation** and **Portfolio Page modules**. This focused approach allows for a deep dive into the critical functionalities associated with a user's ability to create, customize, and manage their digital portfolio. The testing strategy encompasses various levels and types of tests, as outlined below, to ensure the application meets its specified functional and, where applicable, non-functional requirements within the in-scope areas. The grouping under supplemental functional and non-functional tests reflects the current project's needs and resource allocation.

Test	Primary Contact	Comments
Required Functional Testing levels:		
Unit	Development Team	Each individual component and function will be tested in isolation during development to ensure correctness of code logic

System/ Integration	QA Team	End-to-end testing of workflows within the portfolio creation wizard (e.g., navigation between Basic Info, Contact, Content, Image, Theme pages) and data flow to the Portfolio Page.
User Acceptance	Business Analyst	Final validation by key stakeholders and product owners to ensure the Portfolio Creation and Portfolio Page functionalities align with business needs and user expectations.
Production Verification	QA Lead	Quick checks performed after deployment to the production environment to confirm the stability and basic functionality of the Portfolio Creation and Portfolio Page.
Supplemental Functional Tests:		
Security Testing	QA Team	Ensure user data is protected and access is properly controlled.
Usability Testing	QA Team	Conducted to check if the Portfolio Creation flow and Portfolio Page are intuitive, easy to navigate, and provide a satisfactory user experience (within the scope of UI/UX).

Data Verification and Application Conformity Testing	QA Team	Validates that data entered during portfolio creation is correctly displayed and stored on the Portfolio Page, ensuring accuracy and consistency.
Beta Testing	Product Team	Testing by a small group of real users before full release.

4.2 In Scope

This section outlines the overall scope of testing activities specifically for the PortfolioPro application's Portfolio Creation and Portfolio Page functionalities. It details the specific modules, types of testing, and other relevant items that will be covered during this dedicated QA process.

4.2.1 Program/Application / Module

Program: PortfolioPro – A web-based application designed to facilitate the creation, management, and sharing of professional digital portfolios.

Application: Web-based application accessible via desktop and mobile browsers (responsive design).
Modules in Scope:

- **Portfolio Creation:**
  - Basic Information Page
  - Contact Page
  - Content Page
  - Image Page
  - Theme Page
- **Portfolio Page :** This includes the display of the created portfolio and its associated actions (Edit, Copy Link, Share).

4.2.2 Type of Testing

The following types of testing will be conducted within the defined scope for PortfolioPro:

- **Functional Testing:** To validate all user flows, business logic, data input, and outputs within the Portfolio Creation wizard and the Portfolio Page. This includes positive, negative, and boundary testing of all fields and actions.
- **User Interface (UI) Testing:** To ensure the visual appearance, layout, and interactive elements of the Portfolio Creation pages and the Portfolio Page are rendered correctly, are consistent with design specifications, and respond as expected.
- **System Integration Testing:** To ensure seamless data flow and transitions between the various sub-modules within the Portfolio Creation process (e.g., Basic Info to Contact Page) and the successful redirection to the Portfolio Page upon creation.

- **User Acceptance Testing (UAT):** Final validation by business stakeholders to confirm that the implemented Portfolio Creation and Portfolio Page features meet the defined business requirements and user needs.
- **Data Validation and Conformity Testing:** To verify the accuracy and consistency of data as it is entered, processed, stored, and displayed from the Portfolio Creation flow to the final Portfolio Page.

4.2.3 Other In-Scope Items

- **Cross-browser compatibility (basic checks):** Ensuring core functionalities of Portfolio Creation and Portfolio Page work on major modern web browsers (e.g., Chrome, Edge).

- **Responsive Design Checks:** Verification that the in-scope pages adapt correctly to different screen sizes and orientations for mobile and tablet views.
- **Error Message Validation:** Confirmation that appropriate and user-friendly error messages are displayed for invalid inputs or failed operations within the in-scope modules.
- **Accessibility Checks (basic):** Initial checks for basic accessibility considerations like tab order and keyboard navigation within the tested modules.

4.3 Out of scope

This section outlines the areas and types of testing that are explicitly not included in the current testing scope for the PortfolioPro project, based on the current development phase, available resources, and project priorities.

4.3.1 Program/Application / Module

- **Landing Page:** The initial entry point of the application is not part of this specific testing phase.
- **Account Creation Page:** User registration functionalities are excluded from this scope.
- **Login Page:** User authentication and login processes are not covered.
- **Dashboard:** The user dashboard preceding the portfolio creation flow is out of scope.
- **Backend Database Testing (direct):** Direct database testing beyond what is verified through the UI is not included.

4.3.2 Type of Testing

- **Performance Load Testing at Scale:** While basic performance checks may occur incidentally, full-scale load, stress, or endurance testing to determine system capacity under extreme conditions is not part of this phase.
- **Security Penetration Testing:** Deep security vulnerability assessments, penetration testing, or ethical hacking by specialized security experts are outside the scope of this functional testing phase.
- **Automated Regression Testing for Out-of-Scope Modules:** Automated tests will focus only on the in-scope Portfolio Creation and Portfolio Page modules.

- **Internationalization (I18n) and Localization (L10n) Testing:** Multi-language support and regional adaptations are not part of the current feature set or testing cycle.

4.3.3 Other Out-of-Scope Items

- **Third-party Integrations (External to Portfolio Creation/Page):** Any external APIs or services not directly involved in the in-scope modules are excluded.
- **Complex Data Migration Testing:** Scenarios involving large-scale data migration from legacy systems (as there is no legacy system mentioned) are not applicable.
- **Installation Testing:** Testing of the application's installation process on servers or local machines.
- **Recovery Testing:** Testing the system's ability to recover from crashes or failures.

5. Testing Approach & Strategy

The testing approach for the PortfolioPro project robustly follows an **Agile methodology**, where quality assurance is seamlessly integrated into each sprint cycle. The QA team for PortfolioPro collaborates closely with developers and business analysts during critical Agile ceremonies such as story grooming, sprint planning, and daily stand-ups. This proactive engagement ensures a clear understanding of requirements for the Portfolio Creation and Portfolio Page modules and facilitates the early identification and mitigation of potential risks. Acceptance criteria and the provided test scenarios, are meticulously defined for each user story, and comprehensive test cases are created accordingly. Functional, integration, and user acceptance testing specific to the in-scope modules are conducted for every release, with targeted regression testing performed before major deployments to ensure the stability of existing functionalities.

To effectively manage changes in scope or evolving requirements within the Portfolio Creation and Portfolio Page features, the QA team actively participates in backlog refinement and reprioritization discussions. Any instances of scope creep are addressed through thorough impact analysis, leading to updated test planning and resource adjustments. Regular sprint demos showcase the progress of the in-scope modules, and retrospectives provide valuable opportunities to track progress, identify bottlenecks, and continuously improve testing processes. This strategy ensures that testing remains consistently aligned with the business goals and delivery timelines of the PortfolioPro project, while upholding high product quality standards for its core portfolio features.

5.1 Feature Testing Approach & Coverage

The QA team will adhere to a structured and iterative approach to test each new feature and functionality modules of the PortfolioPro application. This ensures that all functionalities are rigorously validated against their defined acceptance. Testing will include:

- **Functional Testing:** Comprehensive validation of all business logic, input validations, auto-population, theme selection, and button functionalities.
- **UI Testing:** Verification of the appearance, responsiveness, and interactivity of all user interface elements.

- **Integration Testing:** Focused on the data flow and navigation between different sub-pages within the Portfolio Creation wizard (e.g., from Basic Information to Contact Page) and the seamless transition to the Portfolio Page.
- **Coverage:** Emphasis on ensuring thorough test coverage across all defined test scenarios and their associated test cases, particularly for positive and negative paths within the in-scope modules.
- **Methodology:** Manual testing will be the primary method for new feature validation in this phase, allowing for exploratory testing and immediate feedback during development sprints.

5.2 Regression Approach & Coverage

Regression testing for the PortfolioPro application will be a continuous activity, conducted in every sprint, to ensure that newly added features, bug fixes, or code changes to the Portfolio Creation and Portfolio Page modules do not negatively impact existing, previously validated functionality.

- **Regression Suite:** The QA team will maintain a focused regression suite primarily covering all critical workflows within Portfolio Creation and Portfolio Page.
- **Maintenance:** As the application’s in-scope features evolve, the regression suite will be regularly updated to reflect changes in functionalities and business rules.
- **Execution:** Regression tests will primarily be executed manually for now. However, test cases identified with automation potential will be prioritized for future automation, aiming to improve efficiency and consistency in subsequent phases.
- **Coverage:** Ensures stability and reliability of the core portfolio functionalities across iterative developments.

5.3 Integration Testing Approach & Coverage

Integration testing for the PortfolioPro application will specifically concentrate on verifying the seamless interaction and data exchange between different interconnected modules and sub-pages within the defined scope of Portfolio Creation and Portfolio Page.

- **Focus Areas:**
    - **Internal Module Integration:** Ensuring correct data flow and functional cohesion between the consecutive pages of the Portfolio Creation wizard (e.g., data entered on "Basic Information" appearing correctly on "Contact" or in the final portfolio).
    - **Workflow Integration:** Verifying that key user journeys that span multiple sub-modules, such as navigating through all creation steps and successfully generating a portfolio, work as expected.
    - **Frontend-Backend Interaction (Implicit):** Confirming that data submitted via the UI is correctly processed by the backend and stored persistently, and that stored data is accurately retrieved and displayed on the Portfolio Page.
- **Coverage:** Test cases will be designed to cover the transitions and data consistency across these interconnected components, ensuring a cohesive user experience.

5.4 Performance & Security Testing Approach & Coverage

For the current testing phase of the PortfolioPro application, performance and security testing will be limited and primarily conducted at a basic level within the scope of Portfolio Creation and Portfolio Page.

- **Performance Checks (Basic):** Focus will be on monitoring the responsiveness and loading times of pages within the portfolio creation flow and the Portfolio Page under normal usage conditions. This is to ensure a reasonable user experience without extensive load generation.
- **Security Checks (Basic):** Basic security considerations will include:
  - **Input Sanitization:** Verification that inputs within portfolio creation fields (e.g., Full Name, Project Name) do not allow for common injection vulnerabilities.
  - **Access Control:** Ensuring that only authenticated users can access and modify their own portfolios.
  - **Data Protection:** Confirming that sensitive data (e.g., email, phone number) entered during creation is handled with basic security practices.
- **Out of Scope:** Advanced load testing, stress testing, comprehensive penetration testing, and deep vulnerability assessments are outside the scope of this focused testing phase and will be handled in later stages or by specialized teams if required.

5.5 Test Automation Approach & Coverage

The automation strategy for the PortfolioPro application will focus on strategically leveraging automation to enhance testing efficiency and reduce manual effort for repetitive, stable test cases within the in-scope modules.

- **Initial Focus:** Initially, manual testing will be crucial for validating new features and complex scenarios within Portfolio Creation and Portfolio Page.
- **Gradual Automation:** Automation will be introduced gradually, targeting high-priority and stable regression test cases that are executed frequently
- **Automation Suite:** The QA team plans to develop an automated regression suite for core functionalities like creating a basic portfolio, navigating key pages, and verifying critical elements on the Portfolio Page.
- **Benefits:** This approach aims to provide faster feedback on code changes, ensure consistent test results, and support smoother integration into potential future CI/CD pipelines for the in-scope features.

5.6 Test Data Management

Effective test data management is fundamental for rigorously validating the features of the Portfolio Creation and Portfolio Page modules in the PortfolioPro application.

- **Data Mix:** The QA team will utilize a combination of realistic scenarios and deliberately crafted dummy data to simulate diverse user interactions during portfolio creation (e.g., varying lengths of descriptions, different types of content for sections).
- **Coverage:** Data sets will be designed to cover various validation types, including positive, negative, and boundary conditions.

- **Privacy:** While PortfolioPro does not involve highly sensitive financial or health data, any personally identifiable information (PII) used in test environments will be managed responsibly.
- **Refresh Strategy:** Test data will be refreshed or reset as needed to ensure consistency across test cycles and to support accurate, repeatable test results, especially for scenarios involving portfolio creation and deletion.

5.7 Testing Phases, Scope, Deliverables, Acceptance Criteria, Impacted Applications

This section outlines the key elements governing the testing efforts for the Portfolio Creation and Portfolio Page modules of the PortfolioPro application.

Testing Phases

- **Unit Testing:** Performed by developers for individual code components within the in-scope modules.
- **System/Integration Testing:** Conducted by the QA team to verify the end-to-end flow and interaction between sub-modules of Portfolio Creation and the Portfolio Page.
- **User Acceptance Testing (UAT):** Final validation by business stakeholders for the in-scope functionalities.
- **Regression Testing:** Iterative testing to ensure no new defects are introduced to the existing features of Portfolio Creation and Portfolio Page.

Scope

- **Functional Requirements:** Validation of all functional requirements detailed in FRD
- **UI/UX Verification:** Coverage of user interface elements, responsiveness, and basic usability within the in-scope modules.
- **Data Integrity:** Verification of data accuracy, consistency, and persistence related to portfolio creation and display.

Deliverables

- **Test Strategy and Test Plan:** This document, defining the overall approach and implementation for testing activities.
- **Test Scenarios:** High-level descriptions of functionalities to be tested.
- **Test Cases:** Detailed steps for execution, expected results, and test data for in-scope functionalities.
- **Test Execution Reports:** Summaries of test execution progress and results for specific cycles or sprints.
- **Defect Reports:** Comprehensive documentation of identified issues, including severity, priority, and status.
- **Requirement Traceability Matrix (RTM):** Linking requirements to test cases for coverage verification.
- **Test Summary Report:** Consolidates outcomes from all testing phases for the in-scope modules, highlighting requirement coverage, defect status, and overall readiness for release.

Acceptance Criteria

- Successful execution and passing rate of all critical and high-priority test cases for Portfolio Creation and Portfolio Page.
- No unresolved critical (P0) or high-priority (P1) defects remaining in the in-scope modules.
- Formal approval and sign-off from business stakeholders for the tested functionalities.
- Compliance with defined functional and basic UI/UX benchmarks.

Impacted Applications

- **PortfolioPro Web Application:** The primary application itself.
- **User's Web Browsers:** Impact on user experience across supported browsers due to UI/functional changes in the in-scope modules.

6. Overall Testing approach

This section outlines the comprehensive approach to testing the PortfolioPro application, particularly focusing on the Portfolio Creation and Portfolio Page modules, ensuring alignment with business requirements, defined quality standards, and Agile delivery timelines throughout the project lifecycle. The strategy emphasizes early involvement of QA, continuous feedback, and iterative testing to build quality into the product from the ground up.

6.1 Test Design Approach

The test design process for PortfolioPro's in-scope modules will be meticulously guided by Agile practices and a clear understanding of the functional requirements. The following key aspects will be considered to ensure effective and comprehensive test coverage:

6.1.1 Design Methodology

- The primary approach for test design will be **Requirements-Based Testing (RBT)**, where test cases are directly derived from the detailed functional specifications provided in the FRD.
- **Scenario-Based Testing:** Test scenarios provide a high-level framework for test case creation.
- **Equivalence Partitioning and Boundary Value Analysis:** These techniques will be applied to design efficient test cases for input fields within the Portfolio Creation pages (e.g., for `Portfolio Title`, `Email`, `Phone`, `Project Description`), ensuring comprehensive coverage with a minimal set of tests.
- **State Transition Testing:** For the multi-step Portfolio Creation wizard, test cases will explicitly cover transitions between pages (e.g., "Previous" and "Next" button functionality between Basic Info, Contact, Content, Image, and Theme pages), ensuring correct state management.

- **Positive and Negative Testing:** Both valid (positive) and invalid/error-inducing (negative) test cases will be designed to verify expected behavior and proper error handling, respectively.

6.1.2 Review Process

All test design artifacts, including detailed test cases and test scenarios, will undergo a rigorous peer review process. This involves:

- **Internal QA Review:** Test cases will be reviewed by other QA team members to ensure completeness, accuracy, clarity, and adherence to established test design principles.
- **Cross-Functional Review:** Relevant test cases and scenarios will be reviewed with development and business analyst teams to ensure alignment with requirements, technical feasibility, and business expectations. This collaboration helps in identifying any ambiguities or missing scenarios early in the cycle.

6.1.3 Entry Criteria for Test Design

For the commencement of test design activities for any feature or sprint within the Portfolio Creation or Portfolio Page modules, the following criteria must be met:

- **Finalized and Approved User Stories/Requirements:** The functional requirements for the specific features must be clearly defined and approved by all stakeholders.

- **Defined Acceptance Criteria:** Specific, measurable, achievable, relevant, and time-bound acceptance criteria must be clearly documented for each user story.

- **Availability of Test Data Strategy:** A clear strategy or initial set of test data for the particular feature must be available.

- **Access to Required Documentation:** Access to the latest FRD, mockups, or design documents pertinent to the features under design.

- **Completion of Design Walkthroughs:** Product design walkthroughs or clarification sessions for the new features should have been completed to ensure a shared understanding.

7. Test Execution Approach

The test execution strategy for the PortfolioPro project will ensure thorough validation of all in-scope systems and interfaces, primarily focusing on the Portfolio Creation and Portfolio Page modules. This will be achieved using a strategic combination of manual and automated testing techniques, executed iteratively within the Agile sprint cycles. The approach aims to provide timely feedback on quality, identify defects early, and confirm that the application functions as designed for its core portfolio features.

7.1 Execution Methodology

- The execution methodology for PortfolioPro testing is designed to maximize coverage and efficiency:

- **Manual Testing:**
  - Manual testing will be extensively utilized for **exploratory testing**, allowing testers to discover unexpected behaviors and usability issues within the Portfolio Creation wizard and the Portfolio Page.
  - It will be the primary method for **ad-hoc testing**, enabling quick checks and immediate feedback on new builds.
  - Crucially, manual testing will be performed for **User Interface (UI) validations**, ensuring the visual accuracy, responsiveness, and aesthetic appeal of all elements across various browsers and devices.
  - New features and complex scenarios that are not yet stable for automation will also be tested manually.
- **Automation:**
  - Automation, will be strategically leveraged for **regression testing** to ensure that new code changes do not negatively impact existing functionalities of Portfolio Creation and Portfolio Page.
  - It will be applied to **repetitive tasks** and stable, high-priority test cases.
  - While not a primary focus for this phase, future API validations for backend services supporting portfolio data could also be automated.
- **Virtualization and Mock Services:**
  - For the current scope of PortfolioPro (focused on Portfolio Creation and Portfolio Page), the use of virtualization or mock services is generally **not anticipated as a primary requirement** as direct integration with a live backend for data persistence is crucial.
  - However, if specific third-party integrations (e.g., social sharing APIs) become a dependency for the in-scope modules and are unstable or unavailable during testing, mock services *could* be considered to unblock testing activities, though this is not a planned approach for the initial phase.

7.2 Test Data Management

- Effective test data management is fundamental for executing robust tests on the PortfolioPro application

- **Data Sourcing:**

  - Test data will be either **created manually** by QA engineers, simulating diverse user inputs for portfolio fields.
  - **Dummy data sets** will be generated to cover specific test scenarios, including positive flows, negative paths (invalid inputs), and edge cases.
- **Data Masking/Compliance:** Given PortfolioPro's nature, direct masking from sensitive production sources is unlikely to be a significant concern for the current in-scope modules, as user data for portfolios is created within the test environment. However, any mock or dummy data will adhere to privacy best practices.
- **Data Refresh and Maintenance:** Test data sets will be regularly refreshed or reset as needed to ensure consistency and prevent data pollution across different test cycles. This is crucial for repeatable test execution, especially for scenarios involving creating new portfolios or modifying existing ones. Specific data sets will be maintained for functional, regression, and UAT cycles to support accurate and reliable results.

7.3 Execution Cycles

- Multiple cycles of test execution will be meticulously planned and executed for the PortfolioPro project, ensuring comprehensive coverage and iterative quality improvement for the Portfolio Creation and Portfolio Page functionalities:

- **Dry Runs (Pre-Execution):**
  - Prior to formal test execution cycles, informal "dry runs" may be conducted for complex or newly designed test cases within the Portfolio Creation flow.
  - This helps in validating the clarity and accuracy of test steps, identifying any environment setup issues, and ensuring testers are familiar with the new features.
- **Formal Feature Testing Cycles (Per Sprint):**
  - For each sprint, formal test execution cycles will be conducted immediately following the deployment of new features or bug fixes to the QA environment.
  - These cycles will focus on executing all new and modified functional test cases relevant to the Portfolio Creation and Portfolio Page modules.
  - Each cycle will include robust **defect logging**, tracking, resolution, and subsequent **revalidation** of fixes.
- **Regression Cycles:**
  - Dedicated regression cycles will be performed at the end of each sprint and before major deployments (e.g., to a Staging/Pre-Production environment).
  - These cycles will primarily focus on executing the established regression suite (comprising critical and high-priority test cases for Portfolio Creation and Portfolio Page) to ensure system stability and prevent regressions.
  - Similar to feature testing, each regression cycle will encompass comprehensive defect logging, resolution, and re-verification activities.
- **User Acceptance Testing (UAT) Cycles:**
  - Formal UAT cycles will be scheduled for business stakeholders to validate the Portfolio Creation and Portfolio Page functionalities in an environment closely resembling production.
  - This cycle emphasizes end-user workflows and business sign-off.
- **Production Verification Testing (PVT):**
  - Quick, critical test cycles performed immediately after deployment to the production environment to confirm the basic operational integrity of the Portfolio Creation and Portfolio Page modules.

8. Tools & Hardware / Software Requirements

This section meticulously outlines all essential tools, software applications, and hardware/devices that are required for the successful planning, execution, and reporting of testing activities for the PortfolioPro project, particularly focusing on the Portfolio Creation and Portfolio Page modules.

8.1 Tools

Testing Type / Phase	Tool	Version	Licensed / Open Source / Cognizant Proprietary
Development Unit Testing	- -	- -	- -
Functional / End to End / Regression / UAT/ Prod	Chrome	137.0.7151.69	Licensed
Test Data Management	Microsoft Excel	250416.0.1873 0.20226	Licensed
Test Automation	Selenium	4.34.0	Licensed

8.2 Test Management Tool

The selected test management tool for the PortfolioPro project will support comprehensive quality assurance activities throughout the testing lifecycle, with a focus on streamlining processes for the Portfolio Creation and Portfolio Page modules. Key features provided by the chosen tool will include:

- **Test Case Repository:** A centralized repository for storing, organizing, and managing all test scenarios and detailed test cases related to PortfolioPro.
- **Traceability and Test Coverage:** Robust capabilities to link test cases directly to specific functional requirements and user stories, enabling clear traceability and ensuring comprehensive requirement coverage.
- **Test Execution:** Functionality to plan, execute, and track the progress of test runs, including assigning test cases to testers and recording results.
- **Test Reports:** Generation of various customizable reports, providing insights into test execution status, defect trends, and overall project quality for the in-scope modules.
- **Defect Tracking:** Integrated defect management capabilities for logging, tracking, prioritizing, and managing the lifecycle of defects identified during testing.

9. Assumption

The successful definition and execution of the test strategy for the PortfolioPro project, are predicated on the following key assumptions:

- **Dedicated Test Environment Availability:** A stable, isolated, and representative testing environment that accurately mirrors the production setup will be readily available and maintained for all phases of testing, allowing for consistent and reliable test execution.
- **Subject Matter Expert (SME) and Product Owner Availability:** Key Subject Matter Experts and Product Owners will be accessible for timely clarification of requirements, resolution of ambiguities related to Portfolio Creation and Portfolio Page functionalities, and active participation in acceptance criteria discussions.
- **Development and Unit Testing Completion:** Each user story and feature, particularly within the Portfolio Creation and Portfolio Page modules, will be fully

developed and thoroughly unit-tested by the development team before being released to the QA team for formal feature testing.

- **Timely Defect Resolution:** It is assumed that all critical and high-priority defects identified within a particular sprint for the in-scope modules will be addressed and fixed within the stipulated timelines of that same sprint. Should this not be feasible, these defects will be formally added back to the product backlog and reprioritized for resolution in subsequent builds or sprints based on their severity and priority.
- **Emergency Build Availability:** In the event of any showstopper defects (P0) or critical production issues discovered post-deployment related to the Portfolio Creation or Portfolio Page, emergency builds or hotfixes will be made available promptly to resolve the issues with minimal downtime.
- **Stable FRD and Design:** The Functional Requirement Document (FRD) for Portfolio Creation and Portfolio Page is considered stable and will not undergo significant changes that would necessitate major re-design of test cases within the current sprint.
- **Required Dependencies:** Any external systems or modules that the Portfolio Creation and Portfolio Page functionalities depend on (e.g., image storage, basic user authentication components) will be stable and available in the test environment.

10. Dependencies

The successful and efficient execution of the test strategy for the PortfolioPro project, specifically for the Portfolio Creation and Portfolio Page modules, is critically dependent on the following factors:

- **Test Environment Availability and Stability:** Reliable and adequately configured test environments will be made available consistently during both onshore and offshore working hours, excluding only planned downtime windows, and will be sufficiently supported by technical teams. This environment must accurately reflect the production setup for the in-scope modules.
- **Integrated Test Environment:** The test environment must be fully integrated with any necessary internal or external components that interact with Portfolio Creation and Portfolio Page functionalities (e.g., image storage, theme rendering engine, backend APIs for data persistence) to facilitate comprehensive Integration Testing and System Testing.
- **Requirements Clarity and Ambiguity Resolution:** All queries, ambiguities, or gaps regarding the functional requirements and acceptance criteria for the Portfolio Creation and Portfolio Page modules must be thoroughly clarified and resolved with Subject Matter Experts (SMEs) and Product Owners before the commencement of test design activities for a given sprint or feature.
- **Timely Build Deliveries:** Stable and testable builds of the PortfolioPro application, specifically including updates to the in-scope modules, must be delivered to the QA environment as per the agreed-upon sprint schedule.
- **Development Team Support:** Prompt support from the development team for defect analysis, bug fixes, and re-deployments is essential for maintaining testing velocity.

11. Defect Management

Defect management is a critical process within the PortfolioPro project, ensuring that all identified issues related to the Portfolio Creation and Portfolio Page modules are systematically tracked, appropriately prioritized, and resolved efficiently to uphold the highest standards of product quality. The following process outlines the structured steps for managing defects throughout the project lifecycle:

11.1 Defect Process Steps

- The defect management process for PortfolioPro aims to streamline the identification, reporting, resolution, and verification of software defects:

- **Initial Validation:**
    - When the QA team identifies a discrepancy between the expected behavior and the actual application build, it is first internally validated.
    - Subsequently, the suspected defect is discussed with the responsible Developer or Subject Matter Expert (SME) to confirm its validity and whether it constitutes a genuine defect. This preliminary step helps to minimize invalid defect entries.
- **Clarification Communication:**
    - Any necessary clarifications or further details regarding the identified issue are exchanged, typically via the project's chosen communication platform (e.g., Microsoft Teams, Slack) or email.
    - For email communication, a standardized subject format will be used.
- **Defect Logging:**
    - Upon confirmation that the issue is a legitimate defect, the QA team logs it in the designated Agile project management and defect tracking tool (e.g., Jira, Azure DevOps, TestRail).
    - Key information captured includes:
        - **Summary/Title:** A concise description of the defect.
        - **Description:** Detailed explanation of the issue, observed behavior, and expected behavior.
        - **Steps to Reproduce:** Clear, step-by-step instructions to reliably replicate the defect.
        - **Environment:** Details of the browser, operating system, and application version where the defect was observed.
        - **Attachments:** Screenshots, video recordings, or relevant log files to aid in understanding.
        - **Status:** Initial status set to `Open` or `New`.
        - **Schedule State:** Typically `Defined` or `To Do`.
- **Severity and Priority Assignment:**
    - The QA team assigns a **Severity Level** based on the impact of the defect on the system's functionality
    - A **Priority Level** is assigned based on the urgency of the fix
- **Developer Assignment:**
    - The newly logged defect is then assigned to the developer primarily responsible for the affected functionality or user story within the Portfolio Creation or Portfolio Page modules.
- **Follow-Up and Communication:**

- The QA team maintains regular follow-up with the assigned developer until the defect is resolved.
- Updates on defect status are shared during daily scrum meetings, and reminders are sent via the project's communication channels if necessary to ensure timely progress.

- **Fix and Deployment:**
  - Once the developer implements a fix for the defect, the updated code is deployed to the QA test environment.
  - The developer updates the defect status to `Fixed` in the defect management tool.
  - A notification is typically sent to the defect submitter (QA) and relevant stakeholders.
- **Verification and Closure:**
  - The QA team verifies the implemented fix by re-executing the steps to reproduce the defect and performing any necessary regression tests.
  - If the defect is successfully resolved and no new issues are introduced, the defect status is changed to `Closed`.
  - If the defect persists or new issues arise due to the fix, the status is reverted to `Reopened`, and the Schedule State is updated (e.g., to `In Progress`), along with re-assignment to the developer with clear comments.

12. Defect Severity & Priority Definition or Classification Guidelines

Defect Severity: This indicates the impact or seriousness of the defect on the functionality of the PortfolioPro application.

Defect Severity	Definition / Classification Guideline
Critical	**Fatal Error or "Show Stopper":** Core functionality of Portfolio Creation or Portfolio Page is completely broken or unavailable, preventing any further progress. No workaround exists. Testing cannot continue in the affected area.
High	**Major Functionality Failure:** Prevents the system from meeting a significant business requirement. A major functionality within Portfolio Creation or Portfolio Page is severely impaired or unusable. A workaround might exist but is complex, difficult, or time-consuming. Testing can continue in other unaffected areas but is significantly slowed down in the impacted module.
Medium	**Affects Functionality (Not Critical):** A non-critical functionality is impaired, or there is a noticeable deviation from the expected behavior. A simple and practical workaround is readily available. Testing can proceed with minor interruptions.
Low	**Cosmetic Issue / Minor UI Glitch:** A minor visual flaw, spelling error, or minor UI inconsistency. Does not affect

	any core functionality or business flow. Testing can proceed without any interruption.

Defect Priority: This indicates the urgency with which a defect needs to be fixed and influences its placement in the development backlog.

- **High:** Must be fixed immediately or in the current sprint. Typically assigned to Critical (P0) and High (P1) severity defects that block critical paths or impact user experience significantly.
- **Medium:** Should be fixed in the current or next sprint. Assigned to Medium (P2) severity defects or High (P1) defects with a reasonable workaround.
- **Low:** Can be fixed in future releases or if time permits. Typically assigned to Low (P3) severity defects or minor improvements.