Project Management Plan (PMP)

Project Name: Travel Advisor Web Application

Date: 7-4-2024

# Purpose

The purpose of this project is to develop a web-based application that allows users to plan their holidays efficiently. It includes features such as registration, login, flight booking, rating experiences, and admin controls. The platform aims to guide users from the start to the end of their travel journey.

# Scope

**In-Scope**

* User registration and login with validation.
* User profile creation and secure authentication.
* Search and explore travel destinations with image gallery.
* Book flight tickets via third-party integration.
* Rating and feedback system.
* Admin dashboard to manage users, comments, and content.
* UI for PC browsers.
* Validation testing
* System design (high level – low level)

**Out of Scope**

* Mobile application development.
* Payment processing integration.
* Hotel bookings.
* Offline or desktop app versions.
* Unit and integration testing

# Roles & Responsibilities (R&R)

| **Role** | **Name** | **Responsibility** |
| --- | --- | --- |
| Project Manager | Mohamed Mohsen | Planning, tracking milestones, and team coordination |
| Developer | Ahmed abdelrahman,  Abdelkerim abdo | Implement features, fix bugs |
| Tester | Caroline, Asmaa, moaaz | Test features, report bugs |
|  |  |  |

# Change requests Management (CR)

1. Submit Change Request Form to Project Manager.
2. PM evaluates impact (timeline, cost, scope).
3. Review with Team.
4. If approved, update PMP and communicate to team.
5. Track in Change Log.

# 5. Project Constraints

- Platform must be web-based and compatible with PCs.  
 - System must support unique user IDs.  
 - Admin features are required to manage users and content.

# 6. Critical Risks

| **Risk** | **Impact Level** | **Mitigation Strategy** |
| --- | --- | --- |
| Scope Creep | High | Strict change control process |
| Delay in third-party API | Medium | Use mock API for early integration testing |
| Security vulnerabilities | High | Apply security best practices & regular audits |
| Lack of test coverage | Medium | Integrate automated testing |
| Stakeholder availability | Low | Schedule meetings in advance |

# 7. Deliverables

- Functional web application with login, booking, gallery, and rating modules.  
 -system design (high level – low level)  
 - Test cases and test report.  
 - Final project report .

# 8. Communication Management

| **Stakeholder** | **Method** | **Frequency** | **Tool Used** |
| --- | --- | --- | --- |
| Team Members | Standup meetings | Twice in a week | Teams / Google Meet |
| Client / PO | Status Updates | Weekly | Email / Google Meet |
| QA & Dev Team | Bug Reporting | As Needed | Trello |

# 9. Main Milestones

• Requirements analysis(SIQ,CRS,SRS) – [two weeks ]  
 •System design (high level- low level) – [one week]  
 • Frontend development – [one week]  
 • Development Phase 1 (Login, Gallery) – [one week]  
 • Development Phase 2 (Booking, Ratings) – [one week]  
 • Testing & QA – [one week]  
 • Final Review & Delivery – [one week]

# 10. Work Breakdown Structure (WBS)

1. Requirements Gathering

- SIQ  
 - CRS  
 2.System design  
 -high level  
 -low level  
 3. Frontend Development   
 4. Backend Development  
 - User Management  
 - Gallery  
 - Booking System  
 - Rating System  
 5. Integration  
 6. Testing  
 -functional  
 -nonfunctional  
 -system testing  
 -acceptance testing  
 7. Deployment  
 8. Documentation

# 11. SDLC Model – Waterfall

The **Waterfall model** is a linear and sequential approach. Each phase must be completed before the next begins.

**Phases of Waterfall for this Project:**

1. **Requirement Gathering**
   * Collect all system and business requirements
   * Freeze scope early
2. **Implementation (Development)**
   * Frontend, backend, and third-party API integration
3. **Testing**
   * Functional testing, bug fixing, regression testing
4. **Deployment**
   * Application is deployed to production
5. **Maintenance**
   * Handle bug reports, update content/features if needed

# 12. Configuration Management

1- **Version Control :** (GitHub)

. **Branch Strategy:** (main) for stable releases, (dev) for ongoing development, (feature) for individual features

. **Backup :** Weekly backups of codebase.

.GitHub Repository:

https: //github.com/Mohamed-Mohsen98/Travel-Advisor-Web-Application/tree/develop

2- Naming Convention Policy:

All configuration items will follow a standardized naming convention to facilitate identification, categorization, and version control.

Format:

(Projectcode)(Module/Feature/D/[version).[docx)

Each Cl within these documents also follows a consistent structure:

[ProjectCode]-(Type]-[Component]-[Sequence)

Examples:

-Document naming:

LH\_TEST\_LOGIN\_001\_v1.0.docx

-Configuration item:

LH-TC-LOGIN-002