Project Management Plan (PMP)

Project Name: Travel Advisor Web Application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change description** | **Reviewed by** |
| 1.0 | 5/4/2025 | Mohamed Mohsen | Initial creation | Asmaa Sayed |
| 1.1 | 5/4/2025 | Mohamed Mohsen | Purpose, scope and roles & responsibilities section are created | Asmaa Sayed |
| 1.2 | 6/4/2025 | Mohamed Mohsen & Asmaa Sayed | Change request section created | Asmaa Sayed |
| 1.3 | 7/4/2025 | Ahmed Abdelrahman | Project constraints section created | Asmaa Sayed |
| 1.4 | 8/4/2025 | Abdelkarim & Ahmed Abdelrahman | Critical risks, Deliverables sections created | Asmaa Sayed |
| 1.5 | 9/4/2025 | Mohamed Mohsen & Caroline Nagy | Communication management & Main milestones | Asmaa Sayed |

# **1.** **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.

# **2.** **Scope**

**In-Scope**

1. User registration and login with validation.
2. User profile creation and secure authentication.
3. Search and explore travel destinations with image gallery.
4. Book flight tickets via third-party integration.
5. Rating and feedback system.
6. Admin dashboard to manage users, comments, and content.
7. UI for PC browsers.
8. Validation testing
9. System design (high level – low level)  
     
    **Out of Scope**
10. Mobile application development.
11. Payment processing integration.
12. Hotel bookings.
13. Offline or desktop app versions.
14. Unit and integration testing

# **3.** **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 |
|  |  |  |

# **4.** **Change requests Management (CR)**

**4.1 Purpose :**

The purpose of this change management process is to manage change requests so that approved changes will be controlled, ensuring the project remains on schedule, within budget and provides the agreed deliverables.

**4.2 Change Control Process :**

**4.2.1 Change Request Initiation**

1. Who can request: (customer - internal team).
2. Where to submit: Submit a formal Change Request (CR) using the Change Request Form via the project management tool [Trello](https://trello.com/b/gRGvAtL0/qa)
3. Form is sent back to the customer after applying impact analysis to decide its status.  
     
   **4.2.2 Impact Analysis**
4. Who performs it: Relevant team leads or subject matter experts (team lead and project managers ).
5. Where: Analysis is documented in the Impact change section in the change request form , stored alongside the Change Request in the shared project repository.
6. Scope of analysis includes:  
      
   1. Timeline impact
   2. Cost/budget impact
   3. Resource allocation
   4. Risk and quality impact  
        
         
        
      **4.2.3 Change Review and Approval**
7. Who approves: Change Control Board (CCB).
8. CCB typically includes:  
      
   1. Customer
   2. Project Manager
9. Where: CCB meetings or via emails .  
     
      
     
   **4.2.4 Change Communication**
10. Who communicates: Project Manager.
11. Where: Through project communication channels — e.g., email updates, project dashboards, or stand-up meetings.
12. What is updated: Project baseline documents, timeline, budget, scope, and risk register.  
       
      
      
       
      
    **4.2.5 Change Implementation**
13. Who implements: Relevant team members (e.g., developers, testers).
14. Where: In the designated environment (e.g., development or testing environments).
15. How it's tracked: Tasks are assigned and tracked in the project tracking tool.  
       
      
      
    **4.2.6 Monitoring and Validation**
16. Who validates: QA or responsible reviewer.
17. Where: Testing/staging environment before final release.
18. How: Post-change validation checklist and sign-off.  
      
    **4.3 Roles and Responsibilities**

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Project manager | Coordinates change process, updates logs, facilitates approval |
| Stakeholders | Submit requests, provide input |
| CCB members | Review and approve/reject changes |
| Team lead | Perform impact analysis |
| QA / Tester | Validate implemented changes |

4.4 **Tools and Templates**

1. **Change Request Form** – Standard form for documenting a change.
2. **Impact Analysis Report Template** – Structure for evaluating potential impact.
3. **Change Log Spreadsheet** – Tracks all changes and their status.
4. **Approval Workflow Tool** – Trello.

# **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. Risk Management**

**6.1 Overview**

The purpose of the risk management process is to proactively manage potential risks throughout the lifecycle of the project. The process defines how risks are identified, assessed, mitigated, and monitored to minimize their impact on the project.

**6.2 Risk Management Process**

The risk management process involves the following steps:

**6.2.1 Risk Identification**

Risks are identified through brainstorming sessions, expert judgment, and team discussions.

Risk identification is conducted at the beginning of each phase and continuously throughout the project.

All team members are encouraged to report any potential risks.

Identified risks are documented in the Risk Register.

**6.3. Risk Assessment**

**Each identified risk is assessed based on two main factors:**

(Likelihood that the risk will occur)

Impact (Effect on project scope, time, cost, or quality if it occurs)

A qualitative assessment is used to categorize risks as Low, Medium, or High for both probability and impact.

The combination of probability and impact determines the priority of the risk.

**6.4. Risk Response Planning**

For each risk, an appropriate response is planned. Responses include:

1. **Mitigation**: Reducing the likelihood or impact of the risk
2. **Avoidance**: Changing the plan to eliminate the risk
3. **Acceptance**: Acknowledging the risk and dealing with it if it occurs
4. **Contingency**: Preparing backup plans in case the risk occurs  
     
      
     
   The selected response is documented in the risk register along with a clear action plan.

**6.5 Risk Monitoring and Review**

1. Risks are reviewed during regular team meetings.
2. New risks are added as they arise.
3. The status of existing risks is updated in the risk register.
4. If a risk is realized, it is tracked as an **issue** and handled through the issue management process.

# **6.6 Critical Risks**

|  |  |  |
| --- | --- | --- |
| **Risk**  Lack of Resources | **Impact Level**  High | **Mitigation Strategy**  Resource Planning |
| 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 unavailability | Low | Schedule meetings in advance |
| Time pressure so multiple people working on the same document at the same time | Medium | Each person working on the document should notify the team of the work he is doing via the team’s communication channel and the document owner has to guarantee that all the work in the document is delivered as notified, once that is done he pauses the work and take a snap of that document and upload it on the repository with the appropriate version. |

# **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**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Name** | **Planned Date from** | **To** | **Status** |
| Requirements analysis(SIQ,CRS,PMP ) | 5 - 4 - 2025 | 12 -4 -2025 | Completed |
| Create SRS , Review PMP | 12 - 4 -2025 | 19 - 4 -2025 |  |
| System design (high level- low level) | 19 - 4 -2025 | 26 - 4 - 2025 |  |
| Development Phase 1 (Login, Gallery) | 26 - 4 - 2025 | 3 - 5 - 2025 |  |
| Development Phase 2 (Booking, Ratings) | 3 - 5 -2025 | 10 - 5 - 2025 |  |
| Testing & QA | 10 - 5 -2025 | 17 - 5 - 2025 |  |
| Final Review & Delivery | 17 - 5 -2025 | 24 - 5- 2025 |  |
| Development Phase 1 (Login, Gallery) | 24 - 5 - 2025 | 31 - 5 - 2025 |  |

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

# **12. Configuration Management**

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

**.Structure:**

📂 PMP

│

├── 📂 Project Management Document

│ ├── 📄 CIL\_Travel\_Advisor\_Web\_Application.docx

│ └── 📄 E\_Travel\_Advisor\_Web\_Application.docx

│

├── 📂 Templates

│ ├── 📄 CRT\_Travel\_Advisor\_Web\_Application.docx

│ ├── 📄 Review\_Templete\_Travel\_Advisor\_Web\_Application.docx

│ └── 📄 PMP\_Travel\_Advisor\_Web\_Application.docx

│

├── 📂 Reviews

│ └── 📄 Review\_Templete\_Travel\_Advisor\_Web\_Application.docx

├── 📂 Requirements

│ ├── 📄 CRS\_Travel\_Advisor\_Web\_Application.docx

│ ├── 📄 SIQ\_Templete\_Travel\_Advisor\_Web\_Application.docx

│ └── 📄 SRS\_Travel\_Advisor\_Web\_Application.docx

├── 📄 .gitattributes

└── 📄 README.md

. **Branch Strategy:** (main) for baseline , (develop) for ongoing development not baseline ,

**.baseline strategy** : project manager will insure that if the document done and have been reviewed , if it done it will upload as a baseline

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

.**Release Strategy :** take a baseline from main branch and that will be a release

every friday of the week

RS\_travil\_advisor\_V01

.GitHub Repository:

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

**2- Naming Convention:**  
 Defining the standard naming conventions to be used throughout the project for documents, files, reviews,deliverables, and versions.

The goal is to keep things consistent, easy to find, and clear for everyone involved in the project.

1. Documents Naming;:

|  |  |  |
| --- | --- | --- |
| Element | Description | Example |
| Initial Letters | Take the first letter of each word in the document name , using uppercase letters. | CRS |
| Project Name | The full name of the project, written in title case with words separated by underscores | Travel\_Advisor\_Web\_Application |

Format

(InitialLetters)\_(ProjectName)

Example

PMP\_Travel\_Advisor\_Web\_Application

1. ID Naming Convention

|  |  |  |
| --- | --- | --- |
| Element | Description | Example |
| Document Type | Indicates the document type | CRS |
| Feature Name | Identifies the specific feature or module to which the requirement belongs | Login |
| Number | Sequential number | 001 |

Format:

(DocumentName)-(FeatureName)-(Number)

Example:

CRS-Login-001

1. Review Documents Naming Convention

|  |  |  |
| --- | --- | --- |
| Element | Description | Example |
| REV | Constant identifier representing review document |  |
| Number | Sequential number | 001 |

Format:

REV-(Number)

Example:

REV-002

1. Trello Naming Convention

|  |  |  |
| --- | --- | --- |
| Element | Description | Example |
| TK | Constant identifier representing trello task |  |
| Number | Sequential number | 001 |
| Task Name | A concise and descriptive title of the task. | Review PMP |

Format:

TK\_(ID)\_(TaskName)

Example: TK\_4\_Reviewing PMP

* In case of multiple people working on the same document being shared online due to time pressure, Each person working on the document should notify the team of the work he is doing via the team’s communication channel and the document owner has to guarantee that all the work in the document is delivered as notified, once that is done he pauses the work and take a snap of that document and upload it on the repository with the appropriate version.

**13.Configuration Item List (CIL)**

* The **Configuration Item List (CIL)** is a structured record that includes all project artifacts and key documents that are subject to **configuration control**. These items are tracked to ensure **version integrity**, **project consistency**, and alignment with the defined project

#### **Purpose:**

To identify, manage, and control project documents that directly impact the project deliverables and outcomes. The CIL serves as a reference for document versioning, status tracking, and audit readiness.

#### **CIL Artifacts Include:**

* PMP\_Travel\_Advisor\_Web\_Application.docx
* Review\_Template\_Travel\_Advisor\_Web\_Application.docx
* CRS\_Travel\_Advisor\_Web\_Application.xlsx
* SIQ\_Travel\_Advisor\_Web\_Application.xlsx
* TC\_Travel\_Advisor\_Web\_Application

**Status Tracking:**

* Each item in the CIL includes a **status field** indicating the release state:
* Example Status: Released
* Status is updated upon PM approval and finalized review.

### **Roles and Responsibilities**

| **Role** | **Responsibility** |
| --- | --- |
| **Project Manager** | Owns the CIL. Ensures all items are properly versioned, reviewed, and released. Coordinates updates and approvals. |
| **Configuration Manager** (if assigned) | Oversees implementation of configuration control processes and tool setup. Maintains CIL accuracy. |
| **Document Owner** (e.g., BA, QA Lead, Developer) | Creates and maintains the content of assigned documents. Submits for review and release. |
| **Quality Assurance** | Ensures that documents meet quality standards before they are marked as released. Participates in document reviews. |

### **14. Review Strategy**

To maintain high quality and accuracy in project deliverables, a structured peer review process is followed. This ensures that each deliverable is evaluated thoroughly before final approval.

#### **(1) Assign Peer Review**

1. **Choose Reviewers**  
    For each task select one or more team members familiar with the task to conduct the review.
2. **Share Materials**  
    Provide the reviewers with all necessary files, documents and access to any related tools or platforms (e.g., Trello, Google Docs).  
     
   **Conduct Review**  
    Reviewers evaluate the deliverable using a standard Review Template, focusing on accuracy, completeness, clarity, and alignment with project requirements.
3. **Collect Feedback**  
    All feedback and comments are documented clearly. The reviewer uploads the review document

#### **(2) Reassign to Owner**

Once the review is complete:

1. **Return Task to Owner**  
    The task is reassigned to the original owner (e.g., Member1), who is notified of the completed review.
2. **Implement Feedback**  
    The owner reviews all comments and makes the necessary changes to address issues and suggestions.
3. **Review Again (If Needed)**  
    After updates are made, the task may go through another round of review to ensure all feedback has been properly addressed.
4. **Final Approval**  
    This cycle continues until the deliverable is fully approved. Once approved, the review task is marked as complete.

**15.Problem Resolution Management**

**1. Problem Lifecycle**

1. **Identify–** Notice and write it down.
2. **Assess –** See how serious it is.
3. **Analyze –** Find out what caused it.
4. **Plan a fix –** Decide how to fix it and who will do it.
5. **Implement** – Do the work to solve it.
6. **Verify –** Make sure it’s really fixed.
7. **Close it –** Write down what we learned**.**

**2. Problem Attributes**

* **ID –** Problem number
* **Title –** Short name
* **Severity –** High, Medium, Low
* **Priority** – How fast it needs to be fixed
* **Owner** – Who is fixing it
* **Status –** New, In Progress, Done

**3. Root Cause**

* **5 Whys –** Keep asking “Why?” until you find the real reason**.**

**4. Problem Status**

* **New –** Just reported
* **Review –** Being assessed
* **Action in Progress –** Work is being done
* **Fixed –** Done.

**5. Estimation**

* **Time –** How long it will take
* **Delay –** Will it slow down the project?
* **Cost** – How much it will cost
* **Risk –** What might go wrong if not fixed?