Life Record Keeper.

Fast Secure Reliable

Requirements Document

Project By: Gulzar Fatima
Course: Capstone - HTTP-5310
Supervisor: Prof. Bernard Monette
Web Development Jan - Dec 2025

Table of Contents

1. Introduction

What is ‘Life Record Keeper’?

Life Record Keeper is a secure web application that helps individuals keep track of important dates and documents related to life records such as work, education, immigration, and travel. It organizes records in one place so they are easy to access and reuse.

Who is it for

- ★ People who struggle to recall dates and details when filling out applications.
- ★ Individuals who want a safe and organized space for their important records.
- ★ Frequent travelers, immigrants, and professionals who often need to submit the same documents and dates for forms, permits, or job applications.

How does it work

When a user signs up, they create a secure account where only they can access their information. From their dashboard, they can:

1. **Create a record** by entering details such as title, category (work, education, travel, immigration), and start/end dates.
2. **Upload supporting documents** (e.g., transcripts, permits, job letters) that are stored securely and linked to the record.
3. **Organize records** with categories, tags, and highlights for quick retrieval.

4. **View records** in a list, timeline, or calendar view for easy navigation.

5. **Export records** into application-ready formats (PDF/CSV) whenever they need to complete forms or submissions.

What does it do

- ★ **Stores** information and documents securely.
- ★ **Organizes** records by categories (work, education, travel, immigration, etc.).
- ★ Provides **one-click export** into application-friendly, easy-to-read formats.
- ★ **Reduces repetitive work and ensures accuracy** when completing applications.

2. Content Evaluation

2.1 Navigation

Global Navigation:

- ★ Logo
- ★ Username/ Settings
- ★ Logout

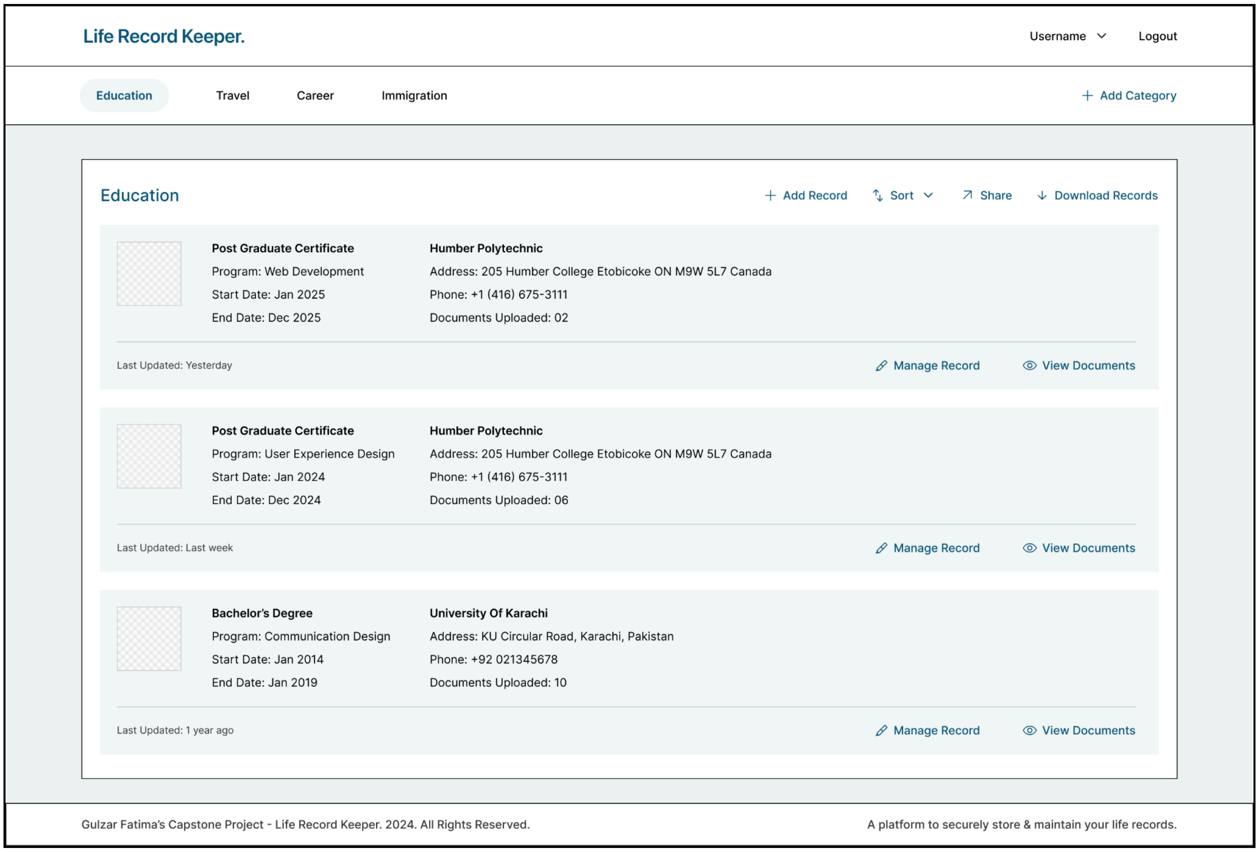
Primary Navigation:

- ★ Categories → Selected category view below
- ★ Add Category → Create category modal

User Flow:

★ Register / Login → Dashboard → Add/Manage Records → Search/ Sort

2.2 Layout, a wireframe at the start



Life Record Keeper.

Username ⌄ Logout

Education Travel Career Immigration + Add Category

Education

+ Add Record ⇅ Sort ⌄ ↗ Share ↓ Download Records

Post Graduate Certificate
Program: Web Development
Start Date: Jan 2025
End Date: Dec 2025

Humber Polytechnic
Address: 205 Humber College Etobicoke ON M9W 5L7 Canada
Phone: +1 (416) 675-3111
Documents Uploaded: 02

Last Updated: Yesterday

Manage Record View Documents

Post Graduate Certificate
Program: User Experience Design
Start Date: Jan 2024
End Date: Dec 2024

Humber Polytechnic
Address: 205 Humber College Etobicoke ON M9W 5L7 Canada
Phone: +1 (416) 675-3111
Documents Uploaded: 06

Last Updated: Last week

Manage Record View Documents

Bachelor's Degree
Program: Communication Design
Start Date: Jan 2014
End Date: Jan 2019

University Of Karachi
Address: KU Circular Road, Karachi, Pakistan
Phone: +92 021345678
Documents Uploaded: 10

Last Updated: 1 year ago

Manage Record View Documents

Gulzar Fatima's Capstone Project - Life Record Keeper. 2024. All Rights Reserved.

A platform to securely store & maintain your life records.

2.3 Content descriptions and sources

The content in this project is made up of user-generated records that are organized into categories such as Education, Travel, and Career. Each record contains descriptive information (e.g. degree name, program name, employer, trip details), important dates (start and end dates), and optional notes. Users can also upload supporting documents such as certificates, transcripts, job letters, or travel confirmations.

All content is entered directly by the user. The system does not rely on any external data sources. The only outside content is limited to general design resources such as icons, open-source libraries, and placeholder images used during development.

2.4 Content Types (Entities)

- ★ User
- ★ Category
- ★ Entry (Record)

3. Functionality & User Stories

3.1 Must-Have Features & User Stories (30)			
#	**Feature**	**User Story**	**Est. Hours**
1	Register Account	As a user, I want to create an account by entering my name, email, and password so that I can have a personal login.	6
2	Email Verification	As a user, I want to verify my email address so I can activate my account securely.	5
3	Login	As a user, I want to log in with my email and password so I can access my personal records.	5
4	Stay Logged In	As a user, I want the option to stay logged in until I log out manually.	3
5	Forgot Password	As a user, I want to reset my password if I forget it.	4
6	Change Password	As a user, I want to change my password while logged in.	4
7	Logout	As a user, I want to log out when I'm done.	2
8	Add Record	As a user, I want to add a new record with title and details.	8

9	Add Dates	As a user, I want to select start and end dates for a record.	4
10	Categorization	As a user, I want to assign categories (work, education, travel, etc.)	4
11	Attach Documents	As a user, I want to upload files (transcripts, permits, job letters) to records.	10
12	Highlight Record	As a user, I want to mark important records (e.g., graduation, visa approval).	3
13	Tag Record	As a user, I want to add custom tags to records	4
14	Edit Record	As a user, I want to update record details.	6
15	Delete Record	As a user, I want to delete a record.	4
16	Delete Confirmation	As a user, I want to confirm before deleting a record.	2
17	List View	As a user, I want to see all records in a list	5
18	Timeline View	As a user, I want to view a timeline of records	8
19	Search Records	As a user, I want to search records by keyword.	6
20	Filter Records	As a user, I want to filter records by category or tag.	5
21	Sort Records	As a user, I want to sort records by newest/oldest.	4
22	Private Records	As a user, I want to see only my own records.	5
23	Secure File Storage	As a user, I want my data to be secure and private.	8
24	Responsive Design	As a user, I want a responsive design that works on mobile.	6
25	Timeline View	As a user, I want to see records in order by date.	3

26	Tagging	As a user, I want to tag records for better filtering.	4
27	Data Privacy	As a user, I want my images to be securely stored.	6
28	Loading States	As a user, I want to see a loading state while data is fetched.	3
29	Accessibility	As a user, I want accessibility-friendly buttons and labels.	6
30	Error Messages	As a user, I want an error message if I enter invalid data.	3
			150-160
3.2 Should-Have Features & User Stories (5)			
#	**Feature**	**User Story**	**Est. Hours**
1	Color-coded Categorization	As a user, I want each category of record to have a unique color.	5
2	Duplicate Record	As a user, I want to copy an existing record.	6
3	Multiple Export Formats	As a user, I want to choose the export format (PDF/CSV/JSON).	10
4	Share (read-only)	As a user, I want to share a read-only view of my timeline	8
5	Calendar / timeline view	As a user, I want to view my records in a calendar grid.	8
			35 - 40
3.3 Nice-to-Have Stories & User Stories (3)			
#	**Feature**	**User Story**	**Est. Hours**
1	AI Integration	As a user, I want the system to suggest records I may have forgotten.	12

2	Animations	As a user, I want animated visualizations of my records.	6
3	Auto-tagging	As a user, I want records to be automatically tagged based on input.	8
4	Verified Access (Future)	As a user, I want trusted organizations to access selected records with my permission.	10
			35 - 40

4. Technical Specifications

4.1 Technology stack

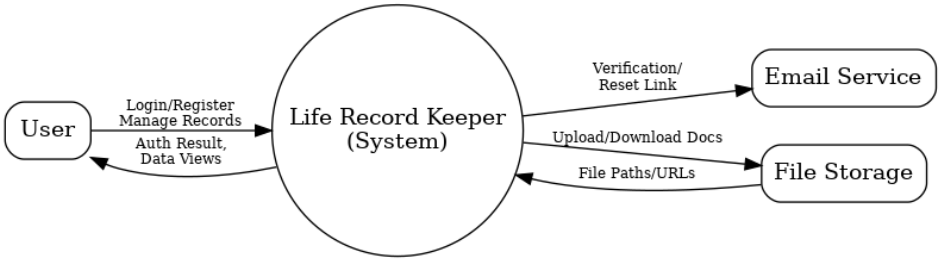
- Frontend: React + Vite
- Backend: Node.js with Express
- Database: MongoDB
- Image Storage: Firebase
- Hosting: Firebase
- Authentication: Firebase
- Design: Figma

4.2 Hosting information

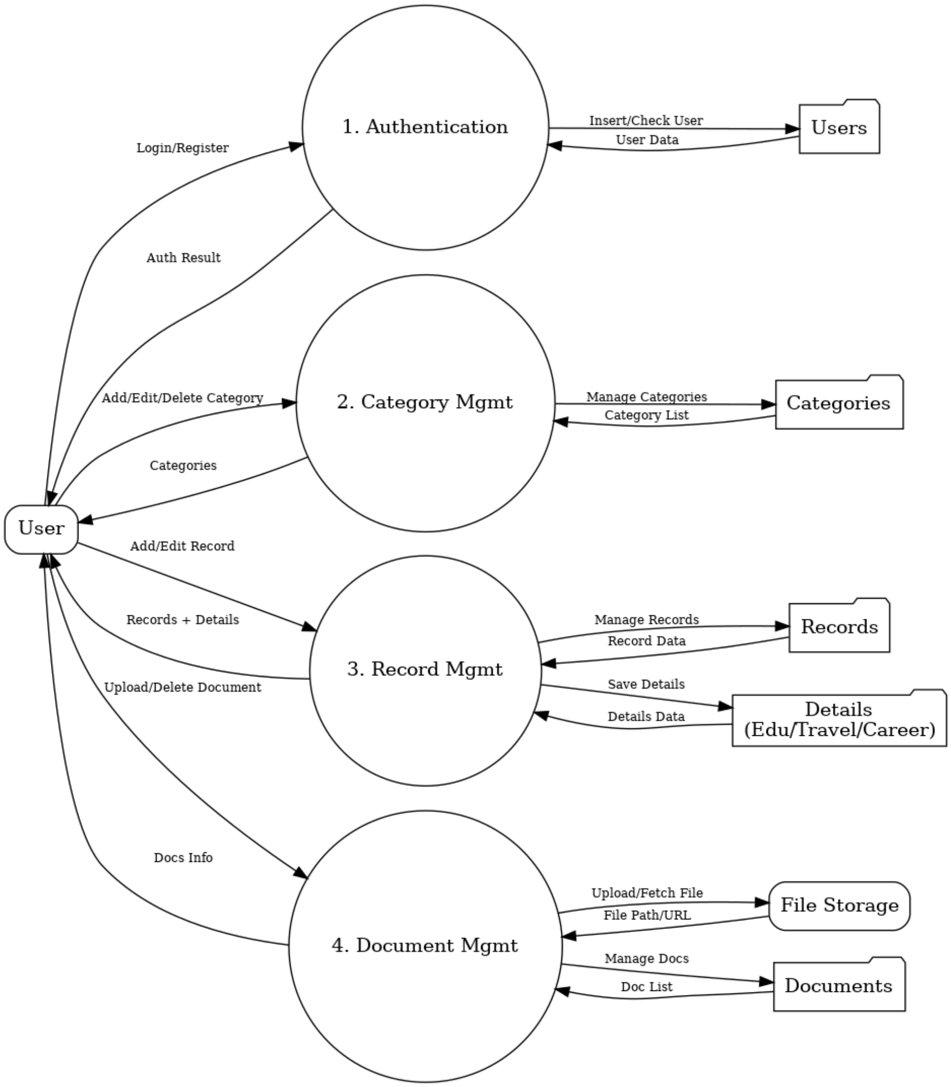
Firebase

5. Data Design

5.1 Data flow diagrams



Login & Record flow



5.2 Database schemas

Entities

Users: accounts that own everything.

Categories: per-user folders/sections.
New users start with Education, Travel, Career (seeded defaults), and you can add more later.

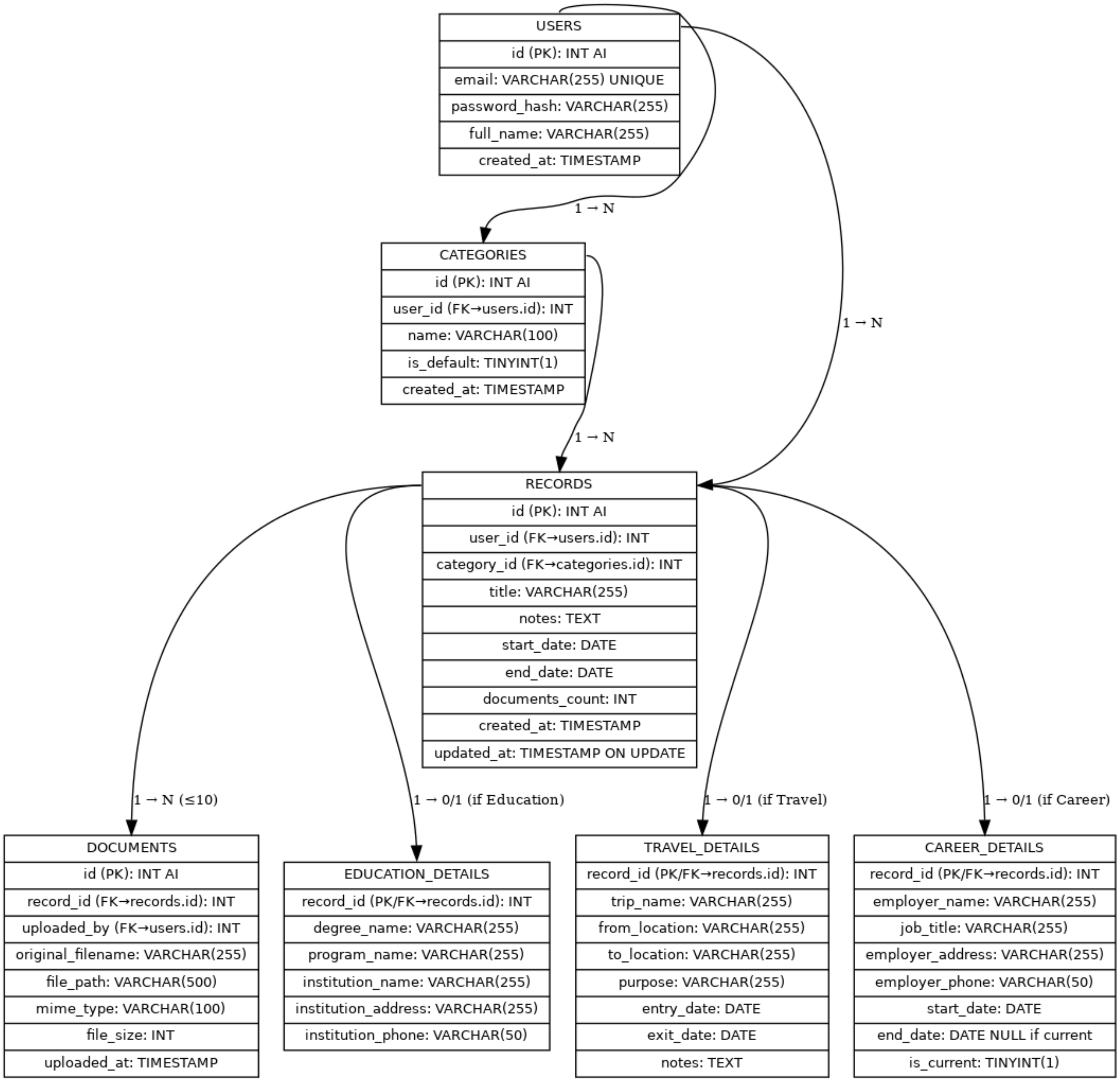
Records: individual entries inside a category (e.g., a degree, a specific job, or a trip). Records carry shared fields (title, notes, start/end dates, last updated, document count).

Documents: file uploads attached to a record (up to 10 per record).

Detail tables: one-to-one tables with fields specific to the category type:

- education_details for Education records,
- travel_details for Travel records,
- career_details for Career records.

ERD



USERS
- id (PK): INT AI
- email: VARCHAR(255) UNIQUE
- password_hash: VARCHAR(255)
- full_name: VARCHAR(255)
- created_at: TIMESTAMP

CATEGORIES
- id (PK): INT AI
- user_id (FK→users.id): INT
- name: VARCHAR(100)
- is_default: TINYINT(1)
- created_at: TIMESTAMP

RECORDS
- id (PK): INT AI
- user_id (FK→users.id): INT
- category_id (FK→categories.id): INT
- title: VARCHAR(255)
- notes: TEXT
- start_date: DATE
- end_date: DATE
- documents_count: INT
- created_at: TIMESTAMP
- updated_at: TIMESTAMP ON UPDATE

DOCUMENTS
- id (PK): INT AI
- record_id (FK→records.id): INT
- uploaded_by (FK→users.id): INT
- original_filename: VARCHAR(255)
- file_path: VARCHAR(500)
- mime_type: VARCHAR(100)
- file_size: INT
- uploaded_at: TIMESTAMP

EDUCATION_DETAILS
- record_id (PK/FK→records.id): INT
- degree_name: VARCHAR(255)
- program_name: VARCHAR(255)
- institution_name: VARCHAR(255)
- institution_address: VARCHAR(255)
- institution_phone: VARCHAR(50)

TRAVEL_DETAILS
- record_id (PK/FK→records.id): INT
- trip_name: VARCHAR(255)
- from_location: VARCHAR(255)
- to_location: VARCHAR(255)
- purpose: VARCHAR(255)
- entry_date: DATE
- exit_date: DATE
- notes: TEXT

CAREER_DETAILS
- record_id (PK/FK→records.id): INT
- employer_name: VARCHAR(255)
- job_title: VARCHAR(255)
- employer_address: VARCHAR(255)
- employer_phone: VARCHAR(50)
- start_date: DATE
- end_date: DATE NULL if current
- is_current: TINYINT(1)

Tables

users

Column	Type	Null	Key	Default	Description	Example Data
id	INT(AI)	NO	PK	-	User_ID	1
email	VARCHAR	NO	UNI	-	Login email	amira@example.com
password_hash	VARCHAR	NO	-	-	Hashed password	!@34%^
full_name	VARCHAR	YES	-	NULL	Display name	Amira Ahmed
created_at	TIMESTAMP	NO	-	CURRENT_TIMESTAMP	Auto	2025-09-06 13:10:00

categories

Column	Type	Null	Key	Default	Description	Example Data
id	INT(AI)	NO	PK	-	Category_ID	10
user_id	INT	NO	FK	-	User	1
name	VARCHAR	NO	-	-	Category name	Education
is_default	-	NO	-	0	1 if seeded default for new user	1
created_at	TIMESTAMP	NO	-	CURRENT_TIMESTAMP	Creation time	2025-09-06 13:10:00

records						
Column	Type	Null	Key	Default	Description	Example Data
id	INT(AI)	NO	PK	-	Record_ID	101
user_id	INT	NO	FK	-	User	1
category_id	INT	NO	FK	-	Category	10(Education)
title	VARCHAR	YES	-	-	Display title	Bachelor of Science
notes	TEXT	YES	-	NULL	Freeform notes	Graduated with honours
start_date	DATE	YES	-	NULL	Start Date	2018-09-01
end_date	DATE	YES	-	NULL	End date	2022-06-15
documents_count	INT	NO	-	0	Count of documents	2
created_at	TIMESTAMP	NO	-	CURRENT_TIMESTAMP	Created timestamp	2025-09-06 13:10:00
updated_at	TIMESTAMP	NO	-	CURRENT_TIMESTAMP on update now()	Auto-updates	2025-09-06 13:10:00

documents						
Column	Type	Null	Key	Default	Description	Example Data
id	INT(AI)	NO	PK	-	Document_ID	5001
record_id	INT	NO	FK	-	Parent record → records.id	101
uploaded_by	INT	NO	FK	-	users.id	1
original_filename	VARCHAR	NO	-	-	Original file name	degree.pdf
file_path	VARCHAR	NO	-	-	Local path or key	/uploads/1/101/degree.pdf
mime_type	VARCHAR	YES	-	NULL	MIME type	application/pdf

file_size	INT	YES	-	NULL	Size in bytes	185432
Uploaded at	TIMESTAMP	NO	-	CURRENT_TIMESTAMP	Upload time	2025-09-06 13:10:00

education_details (1-to-1 with records when category=Education)						
Column	Type	Null	Key	Default	Description	Example Data
record_id	INT	NO	PK/FK	-	records.id	101
degree_name	VARCHAR	YES	-	NULL	Degree	B.Sc. Computer Science
program_name	VARCHAR	YES	-	NULL	Program	Software Engineering
institution_name	VARCHAR	YES	-	NULL	School	State University
institution_address	VARCHAR	YES	-	NULL	School address	123 College Ave, City
institution_phone	VARCHAR	YES	-	NULL	Phone	+1-555-123-4567

travel_details (1-to-1 with records when category=Travel)						
Column	Type	Null	Key	Default	Description	Example Data
record_id	INT	NO	PK/FK	-	records.id	202
trip_name	VARCHAR	YES	-	NULL	title	Toronto trip
from_location	VARCHAR	YES	-	NULL	Origin	London, ON
to_location	VARCHAR	YES	-	NULL	Destination	Toronto, ON
purpose	VARCHAR	YES	-	NULL	Purpose	Conference
entry_date	DATE	YES	-	NULL	Arrival	2023-11-16
exit_date	DATE	YES	-	NULL	Departure	2023-11-23
notes	TEXT	YES	-	NULL	Notes	Visited expo

career_details (1-to-1 with records when category=Career)						
Column	Type	Null	Key	Default	Description	Example Data
record_id	INT	NO	PK/FK	-	records.id	303
employer_name	VARCHAR	YES	-	NULL	Company	Company Ltd.
job_title	VARCHAR	YES	-	NULL	Role	Junior Developer
employer_address	VARCHAR	YES	-	NULL	Address	500 King St, Toronto
employer_phone	VARCHAR	YES	-	NULL	Phone	+1-555-123-4567
start_date	DATE	NO	-	-	Start date of the job	2023-11-16
end_date	DATE	YES	-	NULL	End date (NULL if is_current=1)	2024-10-23
is_current	INT	NO	-	0	1 if current, else 0	1

6. Development Specifications

6.1 Folder structure

/client (React +Vite)

/src/components, /src/pages, /src/routes, /src/styles, /src/lib

/server (Express API)

/src/routes, /src/controllers, /src/models

6.2 File-naming conventions

PascalCase for components, pages and controllers

Example:

LoginForm.jsx

RecordCard.jsx

Kebab-case for route files

Example:

records.routes.js

Overall, ensuring consistent naming conventions to avoid confusion.

6.3 Accessibility standards

- ★ Semantic HTML
- ★ ALT text is provided with images
- ★ Labels and placeholders are used correctly for form fields
- ★ Color contrast is checked
- ★ Design is responsive and usable on different devices

6.4 Responsiveness requirements

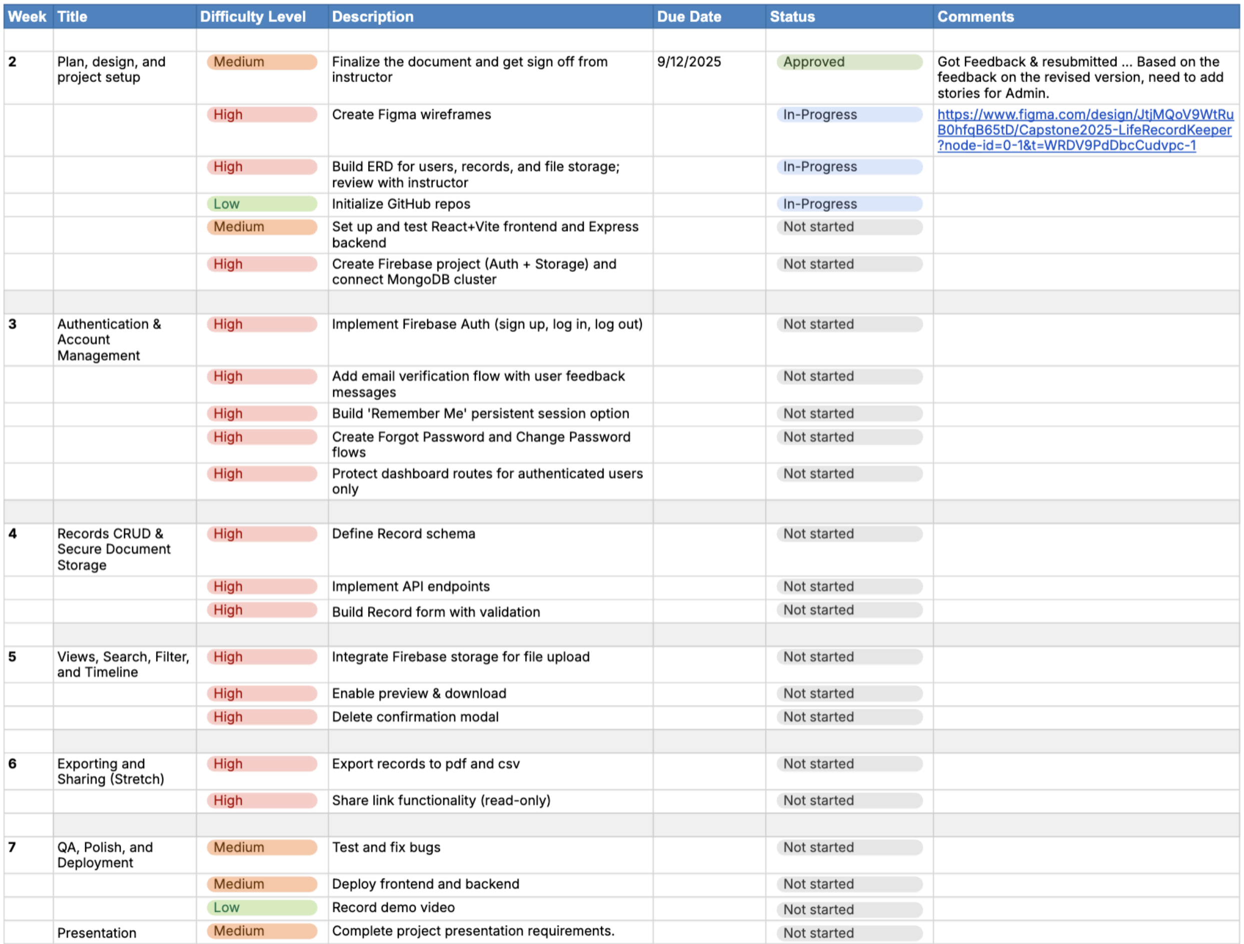
- ★ Application should work efficiently for both mobile and desktop.
- ★ Appropriate use of navigation styles that adapt to different screen sizes.
- ★ Proportionate scale of images and media.
- ★ Flexible layout.

6.5 GitHub links

https://github.com/GulzarFatima/life-record-keeper

7. Schedule

https://docs.google.com/spreadsheets/d/1JM16wOVCFxCD828OZSZopBRY10a9OM84FfDyPlovGxw/edit?usp=sharing



Week	Title	Difficulty Level	Description	Due Date	Status	Comments
2	Plan, design, and project setup	Medium	Finalize the document and get sign off from instructor	9/12/2025	Approved	Got Feedback & resubmitted ... Based on the feedback on the revised version, need to add stories for Admin.
		High	Create Figma wireframes		In-Progress	https://www.figma.com/design/JtjMQoV9WtRuB0hfqB65tD/Capstone2025-LifeRecordKeeper?node-id=0-1&t=WRDV9PdDbcCudvpc-1
		High	Build ERD for users, records, and file storage; review with instructor		In-Progress	
		Low	Initialize GitHub repos		In-Progress	
		Medium	Set up and test React+Vite frontend and Express backend		Not started	
		High	Create Firebase project (Auth + Storage) and connect MongoDB cluster		Not started	
3	Authentication & Account Management	High	Implement Firebase Auth (sign up, log in, log out)		Not started	
		High	Add email verification flow with user feedback messages		Not started	
		High	Build 'Remember Me' persistent session option		Not started	
		High	Create Forgot Password and Change Password flows		Not started	
		High	Protect dashboard routes for authenticated users only		Not started	
4	Records CRUD & Secure Document Storage	High	Define Record schema		Not started	
		High	Implement API endpoints		Not started	
		High	Build Record form with validation		Not started	
5	Views, Search, Filter, and Timeline	High	Integrate Firebase storage for file upload		Not started	
		High	Enable preview & download		Not started	
		High	Delete confirmation modal		Not started	
6	Exporting and Sharing (Stretch)	High	Export records to pdf and csv		Not started	
		High	Share link functionality (read-only)		Not started	
7	QA, Polish, and Deployment	Medium	Test and fix bugs		Not started	
		Medium	Deploy frontend and backend		Not started	
		Low	Record demo video		Not started	
	Presentation	Medium	Complete project presentation requirements.		Not started	

8. Conclusion

Working on this document helped me plan out how the Life Record Keeper system should look and function. I outlined the main features, the database design, the data flow diagrams, and also thought about folder structure, naming, and accessibility. Putting this together gave me a clear roadmap for building the project, and I feel more confident about moving into the development stage.