Capture and Organize

Project members:
FNU Maria – 2873052
Nazia Hassan– 2911095

DEVELOPER GUIDE

1. Overview

Capture and Organize is a fully front-end, browser-based application built using:

- **HTML, JavaScript**
- **Tesseract.js** for OCR
- **Browser APIs**: Notifications, Local Storage

The system follows the **MVC (Model-View-Controller)** architecture for modularity and maintainability.

2. System Architecture

2.1 MVC Breakdown

Model

- Manages application data
- Stores tasks, events, reminders, and settings
- Uses local Storage for persistence

View

- Displays data to the user
- Handles UI rendering and updates

Controller

- Processes user inputs
- Connects actions → logic → UI updates

3. Data Design

3.1 Data Structures

Task Object

```
{
  "id": "t01",
  "title": "Submit project",
  "priority": "High",
  "status": "Incomplete",
  "dueDate": "2025-11-28"
}
```

Event Object

```
{
  "id": "e01",
  "title": "Campus Fair",
  "date": "2025-12-01",
  "image": "base64string",
  "source": "OCR"
}
```

Reminder Object

```
{
  "id": "r01",
  "linkedItemId": "t01",
  "dateTime": "2025-11-28T10:00"
}
```

User Settings Object

```
{
  "theme": "dark",
  "fontSize": "medium"
}
```

3.2 Storage Method

- Uses **local Storage**
- Data stored as JSON strings
- Retrieved and parsed on app load

4. Core Components

4.1 Task Management Component

Handles CRUD operations for tasks.

Key Methods

- addTask()
- updateTask()
- deleteTask()
- loadTasks()
- renderTasks()

4.2 Event Capture & OCR Component

APIs Used

- Tesseract.recognize()

Key Methods

- captureEvent()
- runOCR()
- extractDate()
- saveEvent()

4.3 Calendar Component

Key Methods

- buildCalendar()
- addEvent()
- editEvent()
- deleteEvent()

4.4 Reminder Component

API Used

- Browser Notification API

Key Methods

- scheduleReminder()
- checkReminders()
- triggerNotification()

4.5 Settings Component

Key Methods

- applyTheme()
- applyFontSize()
- saveSettings()
- loadSettings()

Settings persist across sessions.

5. Folder Structure (Recommended)

```
/src
  /css
  /js
    /models
    /views
    /controllers
  /assets
  index.html
```

6. External Interfaces

Camera Interface

navigator.mediaDevices.getUserMedia({ video: true })

OCR Integration

Tesseract.recognize(image, 'png')

Notification API

new Notification ("Reminder Alert!");

7. Testing Procedures

Test Categories

- Unit tests for Models
- OCR accuracy tests using multiple images
- Cross-browser compatibility tests
- Mobile responsive layout tests
- UX and usability tests

8. Limitations

- OCR accuracy may vary depending on image clarity
- Browser storage is limited (5–10MB)
- No cloud sync or user accounts
- Notifications depend on browser support

9. Future Enhancements

- Cloud-based login and data sync
- Better AI event extraction
- Sharing events between users
- Multilingual OCR support