To build a project with the features you mentioned, we can break it down into key components. Here's how the homepage and functionality can be structured:

**Homepage**

The homepage will have a menu with the following elements:

1. **Show Events for Today**
   * Fetch and display events from the database where the date is today's date.
   * Provide an option for each event to generate a greeting card.
2. **Show Events for This Month**
   * Fetch and display all events from the database where the date falls within the current month.
   * Similar to the "today" option, include a feature to generate greeting cards for each event.
3. **Add an Event**
   * A form to input event details (e.g., name, date, description, etc.).
   * Save the event to the database.
4. **Generate a Greeting Card**
   * For each event, generate a greeting card with the theme and text based on event details.
   * Use pre-designed templates or a custom design generator.
5. **Exit**
   * Close or redirect the application.

**Database Design**

A simple relational database table might look like this:

**events Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| id | Integer | Primary key, unique identifier for events. |
| name | Text | Name of the event. |
| date | Date | Date of the event. |
| description | Text | A short description of the event. |
| created\_at | Timestamp | Timestamp for when the event was created. |

**Workflow**

**1. Show Events for Today**

* Query the database:
* SELECT \* FROM events WHERE date = CURRENT\_DATE;
* Display the results in a user-friendly format.

**2. Show Events for This Month**

* Query the database:
* SELECT \* FROM events WHERE EXTRACT(MONTH FROM date) = EXTRACT(MONTH FROM CURRENT\_DATE);
* Display the results grouped by week or day for better readability.

**3. Add an Event**

* Create a form with inputs for name, date, and description.
* On form submission, save data to the database:
* INSERT INTO events (name, date, description, created\_at)
* VALUES ('Event Name', 'YYYY-MM-DD', 'Description', CURRENT\_TIMESTAMP);

**4. Generate a Greeting Card**

* Use the event's name and description to create a card.
* Example greeting card layout:
  + Title: Event Name
  + Body: Description + Custom Wishes (e.g., "Have a wonderful day celebrating [Event Name]!").
* Optionally integrate a library to generate PDF or image-based cards.

**5. Exit**

* A simple button to close the application or end the session.

**Technology Stack**

**Frontend:**

* **HTML/CSS/JavaScript** for a dynamic and interactive UI.
* Use frameworks like **React**, **Angular**, or **Vue.js** if building a more advanced UI.

**Backend:**

* **Node.js**, **Django**, **Flask**, or **PHP** to handle database interactions and logic.

**Database:**

* **SQLite** (lightweight), **MySQL**, or **PostgreSQL** for storing events.

**Optional Tools:**

* **Figma** or **Canva API** for designing greeting cards.
* **Python Imaging Library (PIL)** or **HTML to PDF libraries** for generating cards programmatically.