RSVP Event App Documentation

By sharinsathiq

---

1. Project Overview

The **RSVP Event App** is a web application built with **ASP.NET Core MVC** that allows users to register for an event by submitting their **name, email, phone number, and photo**. After registration, users receive a personalized event pass with their details and photo, which they can download as an image. The app also features an admin view to see all attendees and download their details as a CSV file.

This project was created to practice full-stack web development, including database integration, file uploads, image processing, and front-end styling using Bootstrap.

---

2. Key Features

- **User Registration:** Attendees fill a form with personal info and upload a photo.
- **Photo Upload:** The uploaded photo is saved on the server and displayed on the pass.
- **Event Pass:** Generates a pass with user details and photo that can be downloaded as a PNG image.
- **Attendee List:** Admin view to list all registered attendees with their photos.
- **Export Data:** Download the attendee list as a CSV file for record-keeping.
- **Responsive UI:** Uses Bootstrap 5 for a clean and mobile-friendly interface.

---

3. Technology Stack

Component	Technology/Library
Backend Framework	ASP.NET Core 8 MVC
Database	SQL Server with Entity Framework Core
Frontend Styling	Bootstrap 5
Image Processing	System.Drawing (for pass image)
File Storage	wwwroot/uploads folder

---

4. Project Structure

```
/Controllers
  - RsvpController.cs  (Handles all request logic)
/Models
  - Attendee.cs        (Data model representing attendee)
/Views
  /Rsvp
    - Form.cshtml        (RSVP form view)
    - Pass.cshtml        (Event pass display view)
    - AttendeeList.cshtml (Attendee list view)
/wwwroot/uploads         (Stores uploaded attendee photos)
appsettings.json         (Database connection string and config)
Program.cs               (Application startup and routing)
```

5. Step-by-Step Workflow

5.1 User Registration

- User visits /Rsvp/Form.
- Fills name, email, phone, and uploads a photo.
- On submission, photo is saved in /wwwroot/uploads and form data saved in SQL Server database.
- User is redirected to the /Rsvp/Pass/{id} page.

5.2 Event Pass Generation

- On /Rsvp/Pass/{id}, user sees their event pass with their details and photo.
- User can download the pass as a PNG image from this page.

5.3 Admin Attendee List

- Admin visits /Rsvp/AttendeeList.
- Sees a table of all registered attendees, including photos.
- Can download the full attendee list as a CSV file by clicking the button.

6. Code Highlights

6.1 Saving Uploaded Photos

- Photos are saved uniquely using a GUID filename inside /wwwroot/uploads.
- File path stored in the database for display.

6.2 Generating Pass as Image

- Uses System.Drawing to create a bitmap image.
- Draws the attendee's photo and text details.
- Returns the image as a downloadable PNG file.

6.3 CSV Export

- Builds CSV content dynamically from database attendees.
- Properly escapes commas and quotes.
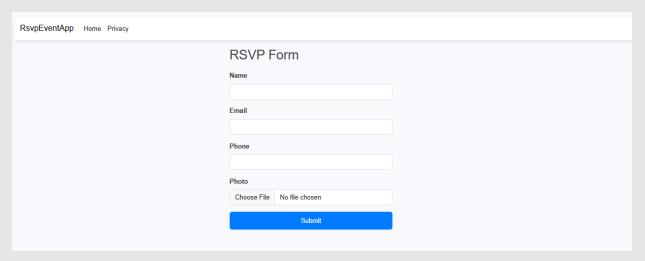- Returns CSV file to the browser for download.

---

7. Running the Project Locally
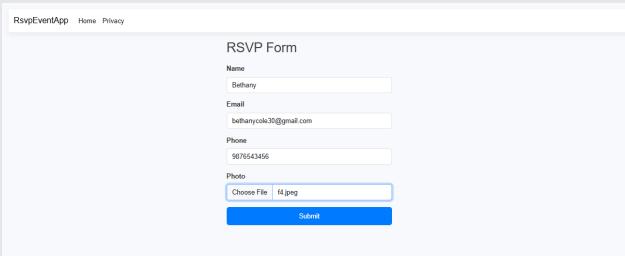
1. Clone the repository.
2. Update connection string in appsettings.json for your SQL Server.
3. Run migrations to create database tables.
4. Run the project with dotnet run or via Visual Studio.
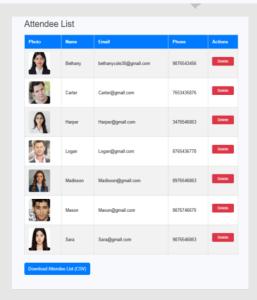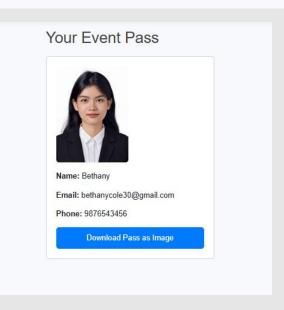5. Access the form at /Rsvp/Form.

---

8. Possible Improvements

- Add authentication to protect attendee list.
- Include QR code on event passes.
- Allow editing or deleting attendee info.
- Use a JavaScript library for improved UI and validations.
- Add pagination or search on attendee list.

---

9. Output Images

10. Conclusion

This project demonstrates:

- Full-stack development skills in ASP.NET Core.
- Handling file uploads and database operations.
- Dynamic image creation in backend.
- Responsive UI with Bootstrap.
- Exporting data in user-friendly formats.

It is a solid foundation for event management or registration apps, and a great learning tool for web development fundamentals.