Interdisciplinary Journal of Virtual Learning in Medical Sciences

Software

# CertificateSight: An Open-Source Automated Certificate Generation and Email Distribution Tool

Himel Mondal[1]*

[1]Department of Physiology, All India Institute of Medical Sciences, Deoghar, Jharkhand, India

## ABSTRACT

Timely issuance of participation certificates for webinars, workshops, and academic events can create a repetitive administrative workload, particularly when participant records and certificate files must be processed manually. CertificateSight is a free, open-source R Shiny application to automate batch generation and distribution of personalized participation certificates while keeping participant data and generated files in the user's local computing environment. The application imports participant information from Microsoft Excel, accepts a user-provided certificate background image, provides an interactive preview, overlays participant names with selectable fonts and font sizes, and generates individualized PDF certificates. An optional email-distribution module sends the generated PDFs through Gmail using Simple Mail Transfer Protocol (SMTP) authentication; internet access is required only for this distribution step. The software was functionally tested with batches of up to 250 certificates across five certificate designs and was also tested by five external users. Under the reported test conditions, the core generation workflow completed successfully. CertificateSight is intended to occupy a specific niche between manual certificate production and cloud-based credential-management platforms: it combines local-first processing, open-source availability, user-supplied templates, and offline PDF generation. The current evaluation does not establish broad scalability, usability, security, or reliability across diverse environments; these aspects require further systematic evaluation.

**Keywords:** Software; Office Automation; Electronic Mail; User-Computer Interface; Certification; R Shiny; SMTP

*Corresponding author:
Himel Mondal,
Department of Physiology,
All India Institute of Medical Sciences, Deoghar, Jharkhand, India.
**Tel:** +919830293119
**Email:** himelmkcg@gmail.com

*Please cite this paper as:*
Mondal H. CertificateSight: An Open-Source Automated Certificate Generation and Email Distribution Tool. Interdiscip J Virtual Learn Med Sci. 2026;17(3):2-9. doi: 10.30476/ijvlms.2026.112257.13923.

Received: 15-07-2026
Revised: 11-08-2026
Accepted: 17-08-2026

## Introduction

Webinars and other online academic and professional-development activities are now widely used for education and knowledge sharing (1, 2). Issuing participation certificates is a recurring administrative task associated with these activities.

During many webinars, the chat section may include repeated questions about when participation certificates will be available. After the event, organizers may also receive follow-up inquiries through email or messaging platforms, particularly when certificates are delayed.

Several approaches are currently available for issuing participation certificates. Automated certificate issuance is available to eligible hosts through Zoom Webinars Plus

*Interdisciplinary Journal of Virtual Learning in Medical Sciences (IJVLMS) is licensed under a Creative Commons Attribution-NoDerivatives 4.0 International License. https://creativecommons.org/licenses/by-nd/4.0*

and Zoom Events (3).

Certifier and Accredible offer cloud-based platforms for issuing digital certificates. Certifier offers a free plan with an annual issuing limit (4). Accredible's plans are based on the number of unique recipients per year (5).

Google Workspace add-ons can automate the generation and email distribution of personalized certificates. Certify'em uses Google Forms responses, while Autocrat merges data from Google Sheets into a document template and can email the resulting files as attachments (6). Both workflows use cloud services and third-party add-ons.

When certificates are prepared individually, organizers must personalize and distribute a separate file for each participant. In many institutions, organizers continue to generate personalized PDF certificates manually using graphic design software, followed by individual or batch email distribution. While this approach requires no specialized platform, it is time-consuming, labor-intensive, and prone to delays and human error, particularly for large events. This delay can increase the administrative workload of organizers and delay certificate delivery. To address these limitations, a free, open-source application was developed to automate these steps while allowing organizers to process participant data locally and generate certificates without a subscription.

## *Design Rationale and Distinguishing Characteristics*

Unlike many existing certificate-generation platforms that rely on cloud-based processing or subscription-based services, CertificateSight was designed as a local, open-source application. Certificate generation, certificate rendering, and PDF generation occur on the user's computer, allowing certificates to be generated without an internet connection after the software environment has been installed. This emphasis on offline operation and user control of data is consistent with principles discussed in the local-first software literature (7). Processing certificates locally reduces the need to transfer participant data to an external certificate-generation service. The application's source code is openly available, allowing users and institutions to inspect, modify, and locally deploy the software without subscription fees. CertificateSight also allows users to upload their own certificate backgrounds and customize typography, rather than restricting certificate generation to predefined templates. Automated email distribution is provided as an optional function through Gmail's authenticated Simple Mail Transfer Protocol (SMTP) service. Unlike local certificate generation, this function requires an internet connection and transmits the relevant recipient information and certificate through the configured email service.

## *Software Architecture and Implementation*

CertificateSight was designed as a local-first R Shiny application comprising four functional components: input management, certificate processing, local output generation, and optional email distribution (Figure 1). The input module accepts a Microsoft Excel workbook with participant details and a JPG or PNG image supplied by the user for the certificate background. The participant data are processed locally, and the certificate-processing module combines the participant's name with the uploaded background using the selected font and font size. An interactive preview allows the user to verify the certificate layout before batch processing.

Following layout verification, the application generates an individualized PDF certificate for each participant. The generated certificates can be saved locally on the user's computer or, when email distribution

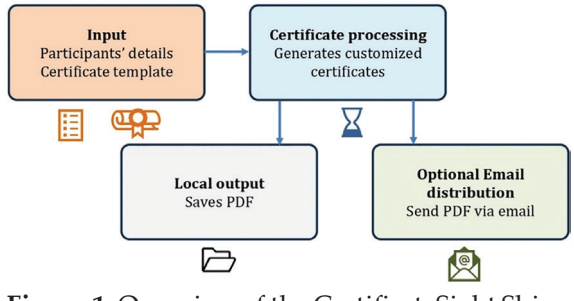


**Figure 1:** Overview of the CertificateSight Shiny application workflow

is selected, attached to personalized emails and transmitted through Gmail's SMTP server. The two output pathways operate independently. Certificates can be generated and saved locally as PDFs without internet access, while automated email delivery requires an internet connection.

### Implementation

CertificateSight was implemented using R and the Shiny framework (8). The application uses the readxl package to import participant information from Excel workbooks (9) and the dplyr package to manipulate and prepare the data (10). It uses the magick package to process images and render participant names on certificate backgrounds (11). The bslib package supports the design of the user interface (12), and the emayili package handles email composition and SMTP communication (13).

Before launch, CertificateSight checks the user's R environment for the required packages and automatically installs any missing dependencies. During certificate generation, the application processes each participant record, renders the participant's name on the uploaded certificate background, and generates an individual PDF certificate. The application assigns each PDF a participant-specific filename and can save it in a designated local output folder.

For email distribution, CertificateSight connects to Gmail's standard SMTP server (smtp.gmail.com) on port 587 (14). It uses a Gmail App Password in place of the user's regular Google Account password for authentication (15). The application replaces the predefined {Name} placeholder in the email subject line or message body with the participant's name and automatically attaches the corresponding PDF certificate.

### Security and Privacy Considerations

CertificateSight separates local certificate generation from optional email distribution. The application processes participant information, generates certificates, and stores the resulting PDFs locally. These operations can run offline without sending participant information to an external certificate-generation service.

When the user enables automated email distribution, CertificateSight establishes an authenticated SMTP connection to Gmail. The user enters a Gmail App Password in the Shiny password field for the email session. The application passes the App Password directly to Gmail's SMTP server for authentication. The user's regular Google Account password is unnecessary. The App Password is intended solely for authentication and should be kept out of the participant dataset and certificate files. The application does not explicitly save it to a file, database, or persistent configuration.

Users can generate and store certificates locally whether or not they enable email distribution. If they choose to send certificates by email, the application requires an internet connection and transmits each recipient's email information and corresponding PDF certificate through Gmail's infrastructure.

### Functional Testing

CertificateSight was evaluated using participant datasets of varying sizes to assess the major functions of the application, including Excel data import, certificate-template upload, font selection, interactive preview, batch PDF generation, local file storage, email customization, and automated certificate distribution. A batch of 250 personalized certificates was successfully generated locally with CertificateSight. The processing time recorded for each of the five certificate templates is reported in Table 1.

Certificate generation time was measured for five templates that varied in image dimensions, file size, and format. The shortest processing time was 33.16 seconds for the 2000 × 1414-pixel JPG template (620 KB). The longest processing time was 1 minute and 33.14 seconds for the 1023 × 1537-pixel PNG template (1510 KB). The application successfully processed all five templates under the test conditions. The resulting certificates were checked for correct rendering of participant names, successful

**Table 1:** Template characteristics and batch processing times for certificate generation in CertificateSight

| Pixel dimensions | Size (KB) | Format | Processing time (mm:ss:ms) |
|---|---|---|---|
| 2000×1414 | 620 | JPG | 00:33:16 |
| 1200×675 | 222 | JPG | 00:34:41 |
| 1600×900 | 458 | PNG | 00:48:01 |
| 1023×1537 | 1510 | PNG | 01:33:14 |
| 1500×1662 | 866 | PNG | 00:58:07 |

*Processing time represents the time required to generate the certificate batch using each certificate template. Values are presented as hours:minutes:seconds:milliseconds. **JPG:** Joint Photographic Experts Group; **PNG:** Portable Network Graphics.

PDF creation, and appropriate file naming. The email distribution function was also tested using participant-specific information and customizable email content.

Five external users independently tested the full workflow, including input preparation, certificate layout configuration, preview, certificate generation, and email distribution. Their feedback helped identify usability issues and guided refinements to the user interface and workflow.

Because the evaluation focused on routine academic certificate-generation workflows, systematic benchmarking across substantially larger datasets was not performed. Therefore, these findings demonstrate successful operation under the tested conditions but do not establish the upper limit of the application's processing capacity or scalability.

## *Features of Application*

CertificateSight is an open-source R Shiny web application that runs locally on the user's computer and automates the generation and optional email distribution of personalized participation certificates. Users upload an Excel workbook containing participant names and email addresses, along with a certificate background image. The application places each participant's name on the background using the font and font size selected by the user. A live preview displays the sample name *Ahana Aarshi* so users can check the layout before generating certificates.

Users can save the certificates locally as high-resolution PDFs and, if desired, email them directly to participants using a customizable email template. Figure 2 illustrates the application's user interface and workflow, including email setup, file upload and download, live certificate preview, and email customization.

To send certificates by email, users must first configure CertificateSight to use their personal email account (Step 1). Users who only want to generate certificates locally can skip Step 1 and begin with Step 2.

### *Step 1: Generating App Password*

To send certificates through Gmail, CertificateSight uses an App Password. This option requires an eligible Google account with 2-Step Verification enabled (15).

### *Users need to:*

Log in to Google Account and open the Security tab.

Ensure that 2-Step Verification is enabled.

Open 2-Step Verification, scroll to the bottom, and select App Passwords.

Enter a name for the application (e.g., CertificateSight) and click Create.

Google will generate a unique 16-character App Password. Copy this password and keep it saved.

### *Step 2: Running the Software*

Before using CertificateSight, users must install R, RStudio, and Rtools on their computer. These components provide the software environment required to run the application locally.

To use the application, open the complete R Shiny source code (16) in RStudio and run the application. At startup, the dependency manager checks your R environment and installs any missing packages. CertificateSight then opens its user interface in RStudio or in

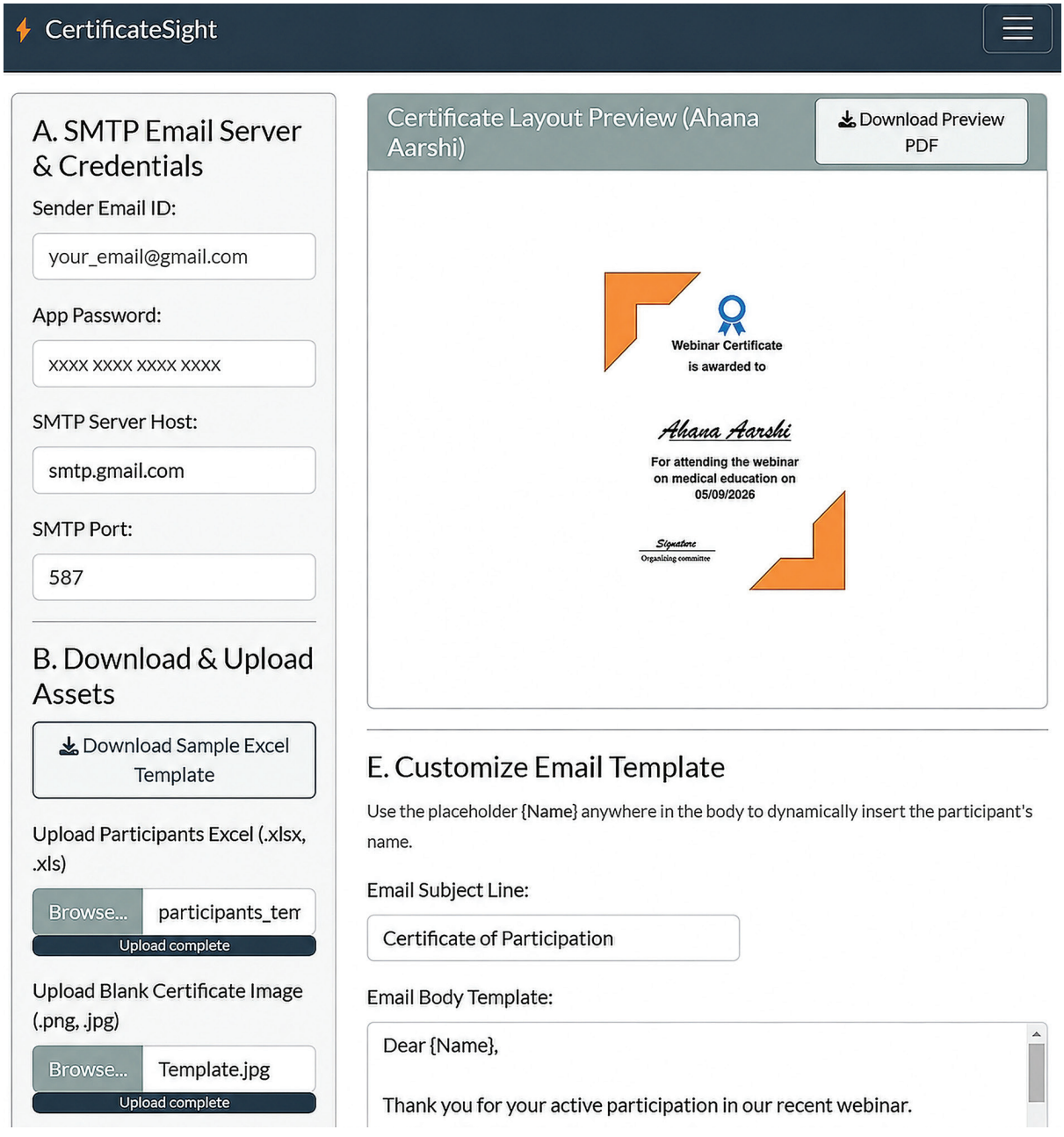


**Figure 2:** User interface of CertificateSight showing key features

your default web browser. Figure 2 shows the resulting interface.

### *Step 3: Generating Certificates*

Once the app interface loads, certificate generation follows a step-by-step workflow.

SMTP Email Server and Credentials

If email distribution is required, enter the Gmail ID and paste the 16-character App Password. If you only want to generate certificates locally, leave these fields blank.

Download and Upload Assets

Click the template button to download the sample Excel workbook. Enter each participant's name and email address under the exact column headings **Name** and **Email**. Choose a blank certificate background with enough space in the center for the participant's name. You can design one using graphic design software or adapt a freely available editable template, such as a PowerPoint certificate template. Upload the completed workbook and the certificate background. Use PNG or JPG format for the background.

### Certificate Typography

Select one of the 12 built-in fonts from the drop-down menu and enter the desired font size in the font size field.

Users can review the live interactive preview box under the "Certificate Layout Preview" window. The part displays a test layout featuring the sample name *Ahana Aarshi* centered on the uploaded design. Use the live preview to adjust the font size until the name fits the layout. To review or print a sample certificate before generating certificates for the full participant list, click "Download Preview PDF".

### Step 4: Generate and/or Dispatch

In the Processing Actions section, choose one of the following options.

### Generate and Email Certificates

You can customize the email subject and body before selecting this option. When you select it, CertificateSight generates a personalized PDF certificate for each participant and sends it to the email address listed in the Excel workbook. The application replaces the {Name} placeholder in the salutation with the corresponding participant's name.

### Generate and Save PDFs Locally

Select this option to save the certificates locally. CertificateSight names each PDF after the corresponding participant and saves it in the "Generated_Certificates" folder within your Documents directory. You can then distribute the saved certificates manually through your preferred communication channel.

### Advantages and Disadvantages

CertificateSight is an open-source application that is free to use and requires no registration or subscription. Users can upload a custom certificate background in JPG or PNG format and incorporate institutional logos or other graphics into the design. The application generates personalized certificates without adding watermarks or branding.

Local processing offers a privacy advantage. Users can generate and save certificates offline without transmitting participant data to a third-party certificate-generation platform. Automated email distribution, however, requires an internet connection and a configured Gmail account. When this option is enabled, CertificateSight transmits each recipient's email address and certificate through Gmail's SMTP service.

**Table 2:** Concise comparison of CertificateSight with Google Forms-based certificate systems

| Feature | Google Forms + Add-ons | CertificateSight |
|---|---|---|
| Cost | Certify'em offers free and paid plans | Free and open source |
| Platform | Cloud-based | Local desktop (R Shiny) |
| Data privacy | Data processed in the cloud | Local during generation. Email delivery transmits recipient data and PDFs through Gmail. |
| Offline use | Not supported | Supported |
| Certificate template | Template selected or supplied by the user | User-uploaded image/template |
| Certificate generation | Online | Online or offline |
| Bulk generation | Supported through add-ons | Supported |
| Customization | Add-on/template dependent | Flexible user-defined customization |
| Email distribution | Supported | Supported via Gmail SMTP |
| Live preview | Depends on the add-on | Interactive live preview |
| Source code | Proprietary | Open source |
| Best suited for | Quick Google Workspace workflows | Privacy-conscious users requiring flexibility and offline capability |

SMTP: Simple Mail Transfer Protocol.

CertificateSight supports both offline certificate generation and optional online distribution. Users can generate personalized PDFs and save them locally for later distribution, even without an internet connection. Local processing keeps participant data and certificate files on the user's computer during generation. Users who choose automated email distribution need a compatible Gmail account, a Gmail App Password, and an active internet connection. When this option is enabled, CertificateSight sends each recipient's email address and corresponding certificate through Gmail's SMTP service.

The current version has two design limitations. It places each participant's name at a fixed position in the center of the page, so users must leave sufficient space at the center of the background. It also lacks a built-in graphics editor. Users must therefore create or edit the background in external design software before uploading it.

Local setup may also pose a challenge for users with limited R experience. CertificateSight runs locally using R and RStudio, and these users may need time to become familiar with the initial setup and launch procedure. Once the application is running, its interface guides users through certificate generation and distribution.

Certify'em and Autocrat both support the generation and email distribution of personalized certificates through Google Workspace (6). Certify'em offers both free and paid plans (17). Table 2 compares these workflows with CertificateSight.

## Conclusion

CertificateSight is an open-source R Shiny application that helps webinar organizers prepare and distribute personalized participation certificates. Its web interface combines R's image-processing capabilities with a workflow that allows users to generate and save PDF certificates locally, even without an internet connection. Educators and academic institutions, including those in resource-constrained settings, can use the application without subscribing to a certificate-generation platform. When an internet connection is available, users can also automate delivery by email. Under the tested conditions, CertificateSight provided a practical workflow with the potential to reduce the administrative effort involved in preparing and distributing certificates.

## Abbreviations

**PDF:** Portable Document Format
**SMTP:** Simple Mail Transfer Protocol
**RStudio:** R Integrated Development Environment
**JPG:** Joint Photographic Experts Group
**PNG:** Portable Network Graphics

## Acknowledgements

The author gratefully acknowledges the developers and contributors of the R programming language and the R Shiny framework, as well as the creators and maintainers of the packages shiny, readxl, magick, emayili, dplyr, bslib, and writexl, whose open-source contributions made the development of CertificateSight possible. The authors also thank Sarika Mondal, Ahana Aarshi, and the other testers for evaluating the software and providing feedback that helped refine its features. Google Gemini Pro was used to troubleshoot and refine portions of the R code during development, and ChatGPT (OpenAI) was used to improve the language and grammar of the manuscript.

## Authors' Contribution

HM conceived the idea, designed and developed the software, wrote the R code, tested and validated the application, analyzed the results, prepared the figures and tables, and drafted, revised, and approved the final manuscript.

## Conflict of Interest

The author declares that there are no conflicts of interest related to this work.

## Funding/Support

This study received no external funding or financial support from any public,

commercial, or non-profit organization.

## Availability of Data and Materials

The data supporting the findings of this study can be found in the article and its cited sources. The R source code for CertificateSight is available at Doi:10.6084/m9.figshare.32988611.

## References


1 Gupta SK, Sengupta N. Webinar as the Future Educational Tool in Higher Education of India: A Survey-Based Study. Technol Knowl Learn. 2021;26(4):1111–30. doi:10.1007/s10758-021-09493-7. PubMed Central PMCID: PMC7811687.

2 Avila CC, Quinn VP, Geiger AM, Kerby TJ, St Charles M, Clough-Gorr KM. Webinar Training: an acceptable, feasible and effective approach for multi-site medical record abstraction: the BOWII experience. BMC Res Notes. 2011;4:430. doi:10.1186/1756-0500-4-430 PubMed PMID: 22013969; PubMed Central PMCID: PMC3213762.

3 Zoom. Using Certification for Zoom Webinars Plus and Events [Internet]. [cited 2026 July 08]. Available from: https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0083983

4 Certifier. Certificate & badge plans [Internet]. [cited 2026 July 10]. Available from: https://certifier.io/pricing

5 Accredible. Accredible pricing [Internet]. [cited 2026 July 10]. Available from: https://www.accredible.com/pricing

6 Sharda P, Bajpai MK. E-certificates generation by using Certify'em, Autocrat and Mail Merge. In: Barman B, Baishya K, editors. Online teaching-learning: issues and challenges. Guwahati: Badan Barman; 2021. p. 1–12.

7 Kleppmann M, Wiggins A, van Hardenberg P, McGranaghan M. Local-first software: you own your data, in spite of the cloud. In: Masuhara H, Petricek T, editors. Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! 2019); 2019 Oct 23–24; Athens, Greece. New York (NY): Association for Computing Machinery; 2019. p. 154–78. doi: 10.1145/3359591.3359737.

8 Kasprzak P, Mitchell L, Kravchuk O, Timmins A. Six years of Shiny in research – collaborative development of web tools in R. R J. 2021;12(2):20–42. doi:10.32614/RJ-2021-004.

9 Wickham H, Bryan J. readxl: Read Excel files [computer software]. Comprehensive R Archive Network. doi:10.32614/CRAN.package.readxl.

10 Wickham H, François R, Henry L, Müller K, Vaughan D. dplyr: A grammar of data manipulation [Computer Software]. Comprehensive R Archive Network. doi:10.32614/CRAN.package.dplyr.

11 Ooms J. magick: Advanced graphics and image-processing in R [Computer Software]. Comprehensive R Archive Network. doi:10.32614/CRAN.package.magick.

12 Sievert C, Cheng J, Aden-Buie G. bslib: Custom 'Bootstrap' 'Sass' themes for 'shiny' and 'rmarkdown' [Computer Software]. Comprehensive R Archive Network. doi:10.32614/CRAN.package.bslib.

13 Collier AB. emayili: Send email messages [Computer Software]. Comprehensive R Archive Network. doi:10.32614/CRAN.package.emayili.

14 Google. IMAP, POP, and SMTP for Gmail [Internet]. [cited 2026 July 12]. Available from: https://developers.google.com/workspace/gmail/imap/imap-smtp

15 Google. Sign in with app passwords [Internet]. [cited 2026 July 14]. Available from: https://support.google.com/accounts/answer/185833

16 Mondal H. CertificateSight: R code [Computer Software]. Figshare. doi:10.6084/m9.figshare.32988611.

17 Certify'em. Plans & pricing [Internet]. [cited 2026 July 15]. Available from: https://www.certifyem.com/pricing