Journal of Network and Information Security
9 (1) 2021, 06-08
http://www.publishingindia.com/jnis

GenC - Student Concession Card Generator

Jagannath Bhat[1*], Abhijith P.[2], Jose Maliyakal[3] and Manish T. I.[4]

[1]UG Student, Department of CSE, Adi Shankara Institute of Engineering and Technology, Kalady, Kerala, India.
Email: jagannathbhat1998@gmail.com

[2]UG Student, Department of CSE, Adi Shankara Institute of Engineering and Technology, Kalady, Kerala, India.
Email: abhijithp64@gmail.com

[3]UG Student, Department of CSE, Adi Shankara Institute of Engineering and Technology, Kalady, Kerala, India.
Email: josemaliyakal98@gmail.com

[4]Professor & Dean, Department of CSE, Adi Shankara Institute of Engineering and Technology, Kalady, Kerala, India.
Email: manish.cs@adishankara.ac.in

*Corresponding Author

Abstract: The software development for the transportation is the key focus in the paper. For students the concession in transportation is provided by the local authorities across the nation. An open source initiative to build software for private bus concession card generation can be a tedious process. Having a software to collect details of students who need the card and generate the cards to be printed in PDF or other supported formats can help speed up the process and make it more efficient. The existing software is a desktop application that runs on outdated operating systems and not compatible with recent configurations and hardware like Microsoft stopped updates to Windows 7 OS, educational institutions need an alternative. These solutions can be used across the institutions for free of cost. Different analytical study can be done from the pattern of the student movements in transportations and future smart transportations can be designed with prior knowledge.

Keywords: **Concession card, Transportation.**

Abbreviations

OS:	Operating System
API:	Application Programming Interface
NSIS:	Nullsoft Scriptable Install System

I. Introduction

Our project aims at providing a cross-platform solution for printing concession cards with very minimal setup. Users can print up to 5 cards per paper. Since it's made of JavaScript, it can run on any modern computer operating system which can run a browser [4].

It has a basic authentication where the user needs to provide a password for using the software. All student details are permanently stored so that they can be reprinted. Storage is handled by a JavaScript library within the software and doesn't need any additional database management software. All data stored are encrypted by default.

For windows, our software is shipped as an installer which can be installed and uninstalled easily. For Linux based OS, it is shipped as a Linux package.

II. Literature Survey

In order to create a solution that works on all popular operating systems, we need a framework for software development that allows compiling the source code to executables that work on all operating systems. There are many technologies available for developing cross-platform desktop apps. The most popular ones are JavaFX and Electron. Based on the comparison done by Alkhars and Mahmoud [1], the Electron framework was chosen for this project. The Electron app gives faster execution time and less memory usage than a JavaFX app.

This project attempts to convert a JavaScript-Based Web Application to a Cross-Platform Desktop Application with Electron as defined by Kitti Kredpattanakul and Yachai Limpiyakorn [2].

In this project, the web application is designed using React. React is an open-source JavaScript library for building user interfaces. The React framework was chosen because it is being built with the intention of being used for building cross-platform native apps as mentioned in [3].

III. Tools/Softwares

Name of Tool	Description
Pouch DB	JavaScript database that stores information as JSON.
Electron	Allows the building of hybrid web applications for desktop operating systems. It combines chromium GUI engine and node runtime.

Name of Tool	Description
React	JavaScript framework for building dynamic web apps.
Node	An asynchronous event-driven JavaScript runtime.
Git	Version Controlling.
NSIS	Nullsoft Scriptable Install System is used for building windows installers.

IV. Implementation

A. Web Application

The project at its core is a web application. The web application is built using an open-source framework called React. In order to create a React-based web application, this project uses a common boilerplate called Create-React-App [9]. In order to use the boilerplate, a JavaScript build toolchain is required. A JavaScript build toolchain typically consists of:

- A package manager, such as Yarn or npm. It lets developers take advantage of a vast ecosystem of third-party packages, and easily install or update them.
- A bundler, such as Webpack or Parcel. It lets developers write modular code and bundle it together into small packages to optimize load time.
- A compiler such as Babel. It lets developers write a modern JavaScript code that still works in older browsers.

The directory structure of the boilerplate looks like the following:

- my-app/
- README.md
- node_modules/
- package.json
- public/
- index.html
- favicon.ico
- src/
- App.css
- App.js
- App.test.js
- index.css
- index.js
- logo.svg

For the project to build, these files must exist with exact filenames:

- ‘public/index.html’ is the page template;
- ‘src/index.js’ is the JavaScript entry point.

The other files can be deleted or renamed. Subdirectories may be created inside ‘src’. For faster rebuilds, only files inside ‘src’ are processed by Webpack. Any JS and CSS files need to be inside ‘src’, otherwise, Webpack won’t see them.

Only files inside the ‘public’ folder can be used from ‘public/index.html’. Top-level directories may be created. They will not be included in the production build [6].

B. API

The project uses ExpressJS to create an Application Programming Interface or API. The API enables the web application to communicate with the database. React is an open-source, front end, JavaScript library for building user interfaces or UI components [8].

Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

The API for this project consists of the following endpoints:

HTTP Method	Path	Description
POST	api/auth	Used for logging in a user. Takes in user and password and returns a JSON Web Token.
POST	api/auth	Used for resetting a user’s password.
GET	api/course	Returns a list of all courses offered by the institution.
POST	api/course	Creates a new course.
PUT	api/course	Modifies a course.
DELETE	api/course	Deletes a course.
POST	api/generate	Returns a PDF file containing concession cards of the provided list of students.
GET	api/student	Returns a list of student records.
POST	api/student	Creates a new student record.
PUT	api/student	Modifies a student record.
DELETE	api/student	Deletes a student record.

C. Database

This project uses PouchDB for storing data locally on the host machine. PouchDB [7] is a JavaScript-based database management system. It stores data in JSON documents. This is very convenient since the project uses JavaScript objects in both the API server and Front-end. PouchDB does not require any additional setup and hence can be installed easily along with the application [10].

V. Result/Conclusion

Successfully developed a software for private bus concession card generation. It runs on any desktop-based operating system that can support a browser [5]. It has a database that doesn't need any additional installation. For Windows operating systems, this software distributed as an installer which also had an uninstaller with it. For Linux based operating systems, this software is distributed as a package.

References

[1] A. Alkhars, and W. Mahmoud, "Cross-platform desktop development (JavaFX vs. Electron)," Dissertation, 2017. [Online]. Available: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1081105&dswid=5131

[2] K. Kredpattanakul, and Y. Limpiyakorn, "Transforming JavaScript-based web application to cross-platform desktop with electron," in K. Kim, and N. Baek (eds.), Information Science and Applications 2018, ICISA 2018, *Lecture Notes in Electrical Engineering*, vol. 514, Springer, Singapore, 2019. [Online]. Available: http://link-springer-com-443.webvpn.fjmu.edu.cn/chapter/10.1007%2F978-981-13-1056-0_56

[3] N. Hansson, and T. Vidhall, 'Effects on performance and usability for cross-platform application development using React Native," Dissertation, 2016. [Online]. Available: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A946127&dswid=-8809

[4] A. Popov, J. Bilokin, T. Solianyk, and K. Vasylchenko, "Development of the system to provide cross-browser compatibility of web application," *2018 IEEE 9th Int. Conf. Dependable Syst., Services and Technologies (DESSERT),* 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8409111

[5] P. Maria, "Javascript beyond the browser," Turun ammattikorkeakoulu, 2018. [Online]. Available: https://www.theseus.fi/handle/10024/151106

[6] J. Justin, and J. Jude, "Go offline," Learn Ionic 2, 2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4842-2617-9_8

[7] R. Cassar, "Abstracting eventually consistent databases for offline web application usage," Dissertations - FacICTCIS - 2016, 2017. [Online]. Available: https://www.um.edu.mt/library/oar/handle/123456789/14734

[8] A. Kumar, and R. K. Singh, "Comparative analysis of AngularJS and ReactJS," *International Journal of Latest Trends in Engineering and Technology*, vol. 7, no. 4. pp. 225-227, 2016. [Online]. Available: https://www.ijrte.org/download/volume-7-issue-4/

[9] S. Aggarwal, "Modern web-development using ReactJS," *International Journal of Recent Research Aspects*, vol. 5, no. 1, pp. 133-137, 2018. [Online]. Available: http://www.divaportal.org/smash/record.jsf?pid=diva2%3A1081105&amp;dswid=5131

[10] R. Cassar, J. Vella, and J. Ellul, "A conflict resolution abstraction layer for eventually consistent databases," *2016 Int. Conf. Eng. and MIS (ICEMIS),* 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7745335