School of Computing and Mathematical Sciences

Cover Sheet for Coursework 1

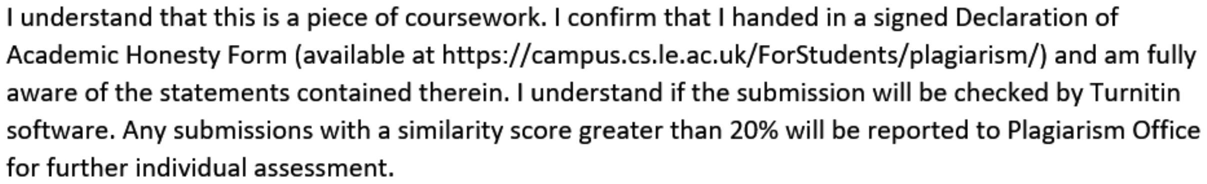
Module Code: co7215

Assignment: Advance Web technology

Surname (in CAPITALS): MARIYAPPAN SUBBURAJ

First name (in CAPITALS): GOBINATH

Student ID: gms27 (239012799)



I understand that this is a piece of coursework. I confirm that I handed in a signed Declaration of Academic Honesty Form (available at https://campus.cs.le.ac.uk/ForStudents/plagiarism/) and am fully aware of the statements contained therein. I understand if the submission will be checked by Turnitin software. Any submissions with a similarity score greater than 20% will be reported to Plagiarism Office for further individual assessment.

Date: 25-11-2023

Signature: Gobinath Mariyappan Subburaj

The Student Course Portal

Desktop Application Development Overview:

The Student Course Portal is a cutting-edge desktop program precisely developed with an advanced technical stack. This application stands out for its novel approach to providing an immersive user experience, leveraging the capabilities of React.js, Electron.js, CSS, and HTML. The strategic selection of Electron.js is critical because it guarantees smooth cross-platform compatibility across Windows, macOS, and Linux platforms. We've achieved the ideal marriage of a native desktop experience and the flexibility and dynamism of web technologies by leveraging Electron.js. This architectural synergy improves the user experience and lays the groundwork for a dynamic, feature-rich, and widely accessible instructional tool.

Programming Languages and Frameworks:

The programming languages and frameworks involved in developing the desktop application are React.js, Electron.js, CSS, and HTML.

React.js: The application's front end is designed with React.js, which provides a responsive and dynamic user interface. The component-based design of React improves code reusability and maintainability.

Electron.js: Electron.js is a framework for developing cross-platform desktop apps that combines Node.js with Chromium, allowing the usage of web capabilities in a desktop environment. This option enables quick development and consistent user experiences across several operating systems.

HTML and CSS: These common web technologies are used to style and structure the program, resulting in a clean and straightforward user experience.

Application Logic and Functionality:

In this application, some of the functionality plays a key role and they are Age Group Filtering, Price Filter, Combination Filter, Dynamic Sorting, and Responsive User Interface.

Age Group Filtering: The program has age group filters to satisfy pupils' various educational demands. Courses are divided into two age groups: 6 to 9 and 10 to 17. This function allows users to easily filter their search depending on the intended age range, ensuring that the courses displayed fit with the kids' individual educational needs and cognitive skills.

Price Filter: Recognising the importance of budget issues in education, the program employs sophisticated pricing filters. Students can filter courses depending on their cost limits, ensuring that the alternatives given are within their financial restraints. The pricing filter capability enables customers to make educated selections, promoting diversity and accessibility in course selection.

Combination Filter: The app's strength comes in its ability to smoothly mix age groups and price criteria. Users may refine their search by defining both the age group and the budgetary limits, resulting in a personalized list of courses that closely meet their parameters. This combination filtering improves the user experience by giving a degree of flexibility that improves the efficiency of the course selection process.

Dynamic Sorting: The program, in addition to filtering, has dynamic sorting options. The course listings can be sorted by relevancy, popularity, or other relevant factors. This dynamic sorting feature offers another degree of flexibility, allowing students to browse courses based on their priorities and interests.

Responsive User Interface: The logic that governs these filtering and sorting algorithms is smoothly incorporated into the React.js framework, resulting in a dynamic and real-time user experience. The program automatically refreshes the shown courses as users interact with the filters and sorting options, offering a fluid and intuitive experience.

This thorough logic not only streamlines the course selection process for students, but also allows them to make educated judgments based on their age group, economic limits, and unique preferences. The Student Course Portal, with its advanced application logic and functionality, exemplifies the dedication to user-centric design and a comprehensive grasp of the different demands of the student group.

Testing and Quality Assurance:

Testing and quality assurance are critical components of the Student Course Portal's development lifecycle, delivering a robust and high-performance product. To evaluate and assure broad compatibility, rigorous testing techniques embracing multiple Windows user situations were used. This thorough testing approach includes unit testing, which examines individual components for correctness; integration testing, which ensures seamless collaboration between different parts of the system; and user acceptance testing, which puts the application in the hands of end users to evaluate its real-world functionality.

Furthermore, the use of Continuous Integration and Continuous Deployment (CI/CD) practices has been a game changer. Automation of testing and deployment procedures not only speeds up development but also reduces the chance of human mistakes by guaranteeing that each code push is subjected to a thorough suite of tests before integration. This dedication to quality assurance not only ensures the application's dependability but also promotes a development culture that prioritizes excellence and customer happiness.

Deployment and Distribution:

For each supported operating system, the program is provided as a stand-alone executable file. Electron.js bundles the program into platform-specific binaries, making it easier for end users to install. This guarantees that students may quickly download and operate the program without having to install any extra prerequisites. To properly manage updates and fixes, the deployment method involves version control.

Security and Data Protection:

Security is a top priority, especially when working with student data. The application adheres to best practices for data security both in transit and at rest. HTTPS is used for secure communication, and sensitive data is secured. To prevent unauthorized access, access restrictions, and authentication measures are in place. Regular security audits and upgrades are part of the development lifecycle to address evolving threats and vulnerabilities.

Case Studies and Examples:

Several case studies and examples were considered during the development of the application. One of the major example is Coursera which mainly focus on professionals and doesn’t have any options for kids to learn coding and robotics. But this application pays way to learn these technologies for this generation.

Conclusion:

The Student Course Portal is a successful desktop program that makes use of new web technologies to give students an efficient and user-friendly experience. The combination of React.js with Electron.js allows for quick development without sacrificing performance or cross-platform compatibility. The extensive testing and security mechanisms put in place ensure the application's dependability and data security.

Future Trends and Challenges:

In the future, we can include a login under one parent and multiple children login which provides both parents and children an enhanced experience of the application without creating multiple accounts. In the current application, one of the major challenges is the connection of the API from the front end to the back end.

Reference:

1) https://react.dev/
2) https://www.electronjs.org/
3) https://www.w3schools.com/html/
4) https://www.w3schools.com/css/
5) https://www.coursera.org/