Politechnika Warszawska
Wydzia Elektroniki i Technik Informacyjnych
Instytut Informatyki

Academic year 2016/2017

Praca dyplomowa magisterska

Dennis Durairaj

Creating PDF Documents from Web Applications

Opiekun pracy:
Tytu Imi i Nazwisko

Ocena .....

.....
Podpis Przewodniczcego
Komisji Egzaminu Dyplomowego

Specjalno: Informatyka –
Inynieria oprogramowania
i systemy informacyjne

Data urodzenia: 1 stycznia 1980 r.

Data rozpoczcia studiw: 1 padziernika 2002 r.

yciorys

Nazywam si ....

.....
podpis studenta

Egzamin dyplomowy

Zoy egzamin dyplomowy w dn. .....

Z wynikiem .....

Oglny wynik studiw .....

Dodatkowe wnioski i uwagi Komisji .....

.....

Streszczenie

Praca ta prezentuje ...

Sowa kluczowe: *sowa kluczowe.*

Abstract

Title: *Creating PDF Documents from Web Applications*

This thesis describes an in-depth research in handling creation of PDF documents through various web technologies in both the client-side as well as server-side.

Key words: *key words.*

Contents

1. Introduction

1.1. PDF - Portable Document Format

PDF was developed in the early 1990s as a way to share computer documents, including text formatting and inline images. In those early years before the rise of the World Wide Web and HTML documents, PDF was popular mainly in desktop publishing workflows.

Why Use PDF?

A properly prepared PDF will maintain the original fonts, images, graphics as well as the exact layout of the file (think of it as an electronic snapshot). A PDF file can be shared, viewed, and printed by anyone using the free Adobe Reader software regardless of the operating system, original design application or fonts. Originally PDF was mostly used by graphic artists, designers and publishers for producing color page proofs. With its evolving technology, however, today PDF is used for virtually any data that needs to be exchanged among applications and users. It is an open file format specification and PDF is available to anyone who wants to develop tools to create, view or manipulate PDF documents.

Why is PDF important?

It is becoming increasingly easy to create PDF files as (from a user's stand-point) the process is almost as simple as printing. Essentially, anything that can be done with a sheet of paper can be done with a PDF. PDF technology is being used more frequently to produce offset printed documents (provided the designer properly embeds fonts and images). Adding to mainstream adoption, of course, is the fact that many applications allow users to save, import or export a document as a PDF (including popular publishing programs like QuarkXPress and CorelDraw), and you can also find a variety of third-party PDF conversion software tools available. With the capability to embed metadata (data about data) in a PDF file, along with the use of security options and electronic signatures PDF is also becoming a standard for data archiving. It may have taken a few years to perfect and years of dedication by the development team at Adobe, but today more and more people are turning to PDF as the solution for something not even thought of in 1993.

Technical details about PDF

A PDF file is basically a 7-bit ASCII file, except for certain elements that may have binary content. A PDF file starts with a header containing the magic number and the version of the format such as PDF-1.7. The format is a subset of a COS ("Carousel" Object Structure) format.[15] A COS tree file consists primarily of objects, of which there are eight types:

— Boolean values, representing true or false

— Numbers

— Strings, enclosed within parentheses ((...)), may contain 8-bit characters.

— Names, starting with a forward slash (/)

— Arrays, ordered collections of objects enclosed within square brackets ([...])

— Dictionaries, collections of objects indexed by Names enclosed within double pointy brackets

— Streams, usually containing large amounts of data, which can be compressed and binary

— The null object

1.2. Web Communication Methodologies

1.3. Current methods to convert data to PDF

2. Creating PDF documents in front-end

2.1. Frontend technologies

Front-end web development (also known as client-side development) is the practice of producing websites or Web Applications using HTML, CSS and JavaScript so that an end user can see and interact with them. The challenges that come with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

2.1.1. HTML5, CSS and JavaScript

What is HTML5?

HTML, or HyperText Markup Language, is the most important element of the World Wide Web. Its the language used to describe what a webpage should look like. However, HTML on its own is pretty boring because it can only deliver static pages; in order to meet the growing demand for more impressive web features, HTML has been coupled with plugins like CSS, Flash, Java, Silverlight, etc.
It has become something of a bloated mess and different browsers implement those features in their own ways. HTML5 is meant to solve HTMLs big problems for a cleaner and more efficient web.

Cascading Stylesheets

CSS is a style language that defines layout of HTML documents. For example, CSS covers fonts, colours, margins, lines, height, width, background images, advanced positions and many other things. Just wait and see!
HTML can be (mis-)used to add layout to websites. But CSS offers more options and is more accurate and sophisticated. CSS is supported by all browsers today.
How do HTML and CSS work together? In general, you use HTML to describe the content of the document, not its style. You use CSS to specify the document's style, not its content.

JavaScript - The language of the Web

JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. From the browser to the server, JavaScript is one of the most versatile and popular languages powering the modern web.

Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.

Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.