

Web-based programming projects

Volume 1 - database applications

Graham Hall

Contents

Volume 1 - database applications

	Forward	
1:	Developing web pages	1
2:	Hardware store	16
3:	Airline booking system	57
4:	Caravan park	117
5:	Historical timeline	185
6:	Delivery planning	271
7:	London Underground	357

Volume 2: graphics applications

Interactive web graphics and computer aided design

Volume 3: computer science applications

Developing on-line programming systems

Forward

This set of books is based on project work carried out over a number of years by Computer Science A-level and Foundation Degree students at Coleg Meirion-Dwyfor in Dolgellau, North Wales. Grateful thanks are due to the students for their enthusiasm and skill in developing design ideas and programs.

A practical programming project is an important component of a course in computing. The project often provides the first realistic opportunity to experience the work undertaken and the challenges faced by computing professionals. Project work should be viewed as an important learning experience, rather than simply as a method of assessment. Students can become highly motivated when working independently on a problem of personal interest over an extended period, allowing them to extend their programming skills and deepen their understanding of computer science.

Many computing projects undertaken by A-level and Foundation Degree students involve the design and development of business and administration systems to run on stand-alone computers, for example: shop sales applications, club membership record keeping, or hotel booking systems. Whilst there is no doubt that such systems are extremely important commercially, they represent only a small proportion of the computing applications which we encounter in our everyday lives at home and at work. The aims of this book are twofold – firstly to encourage students to undertake a wider range of project topics in computing, and secondly to encourage the development of software systems which operate over the internet.

At A-level, a strong case can be made for extending the range of computing projects to topics beyond business data processing:

- Our students combine computing with a wide range of other A-level subjects including: mathematics, electronics, design and technology, sciences and geography. Many students have relatively limited interest in business applications.
- Our students often go on to study Computer Science at University, or to follow courses in new disciplines such as Computer Aided Design or Games Programming. An A-level computing course focussing too heavily on business database applications may be inadequately preparing students for the scientific and mathematical demands of these degree courses.
- The future development of computing is likely to take place in many directions: robotics and control systems, artificial intelligence, environmental modelling, to mention but a few. Students who have focussed on business applications at A-level and then at University may be missing opportunities for careers in these exciting fields.

Developing on-line programming systems

Developing a web-based project rather than a stand-alone application involves a particular set of challenges, but also provides new opportunities. From the point of view of the user, internet sites allow the convenience of on-line shopping, booking tickets and obtaining up-to-date news or travel information, not to mention the value of e-mail and social media communication. Many applications which traditionally might have been distributed on disc can now be used on-line. For example, graphics processing and computer aided design applications can be run on web pages, making them available quickly and conveniently without the need for installation on a local computer.

Web-based projects do, however, require a different methodology. When developing a stand-alone project, a single programming language is generally chosen, for example: Java or Visual Basic. By contrast, web applications often involve a mixture of technologies which can work together on a single web page. In the example projects in this book, we mix HTML, CSS, SQL, PHP, JavaScript and p5.js program code in whatever ways are convenient to achieve the required outcome.

The objective of the series of books is to demonstrate key concepts which can be used as building blocks or components of larger systems. Volume 1 focusses on applications centred around the use of an on-line database. Volume 2 examines interactive graphics applications including computer aided design software. Volume 3 develops computer science applications, including simplified interpreters for programming languages.

Rather than representing finished projects, the example programs form starting points which could be further developed or modified according to the personal interests of the student or the requirements of a client. In each chapter, the programming which is essential to the application is explained in detail, whilst some additional web pages may only be described and illustrated with screen images. The reader may create these additional pages if required, using techniques similar to previous example programs.

In producing a computing project, it is hoped that the student will be assessed mainly on their design skills and technical ability in programming. However, a substantial proportion of the assessment may depend on the production of documentation, and how this aspect is handled must be carefully considered. Mark schemes often sequentially follow the classical stages in systems analysis: identification of a problem, its analysis, the design of a solution, testing and evaluation. In practice, however, software is often developed in a series of cycles. Design discussions take place; programming is carried out; the current software version is demonstrated to the client; then further development ideas are discussed and the process is repeated. A daily development diary is a good way of documenting these software cycles over the extended period of the project. Notes can then be extracted to compile the series of chapters specified by the assessment criteria.

Care must always be taken to meet the Examination Board's requirements. An aspect which can cause particular problems with graphics-based projects is the compiling of adequate evidence of program testing. There is a tendency amongst graphics programmers to work on-screen for long periods, correcting numerous errors in the positioning and movement of lines and shapes until they achieve a fully functional program, but with no hard evidence of the large amount of incidental testing which has been accomplished in the process. Final screen prints of the completed software are unlikely to achieve such high marks for testing as the many pages of alphanumeric data which can be generated from a database. As a way around this problem, students undertaking graphical projects can compile a test log sheet as they work on their software, recording each instance of an error which needs correction. Where possible, screen prints should be made of the incorrect screen display, followed by the corrected screen output. Notes are made on the cause of the problem and the solution which was found. In this way, graphics programmers can generate volumes of test documentation to rival their database colleagues.

Structure of volume 1:

An introduction outlines the programming methodologies which we will use to develop internet-based projects. These include: HTML and style sheets, PHP linked to an SQL database, JavaScript and p5.js processing language. The server requirements for student projects are reviewed, and the use of Unified Modelling Language design diagrams is discussed.

The example projects begin with a website to display a product catalogue for a **hardware store**. The objective is to introduce the use of HTML and style sheets to create and format web pages combining text and photographs, and incorporating a menu system. As with most on-line business applications, data is stored in a database and accessed using program code to carry out *create*, *retrieve*, *update* and *delete* operations. The system used is SQL in combination with the PHP programming language.

In the next example, a simple **airline booking system** is set up using PHP and SQL code. A graphical display using an array of buttons allows customers to select seats on the aircraft. We investigate real-

time methods for temporarily reserving seats to allow the customer to enter contact details and make payment. Seats will be released again if payment is not completed within a specified time.

In the **caravan park** project the use of JavaScript graphics is introduced, allowing a customer to select and book a caravan from a detailed plan of the caravan park. PHP date functions are used to create a booking calendar displaying the weekly availability of each caravan.

The **historical timeline** project develops a simple data retrieval system which can be used to create web pages on historical themes, then compiles information for a visual display. The program introduces aspects of image processing in p5.js.

Delivery planning uses the p5.js programming language for easy manipulation of graphics components on a web page by means of the mouse. The *nearest neighbour* algorithm determines a shortest delivery route between customers.

The **London Underground** program incorporates more complex algorithms in PHP, using arrays of objects to determine possible routes between stations. Underground lines may include loops and multiple branches, and shortest routes may involve one or more changes of train.

Resources

The resources needed to undertake the projects in this book are readily available at reasonable cost or free of charge.

- The first requirement is an Internet Service Provider who can arrange password protected FTP access to a server, along with the use of an SQL database. Many providers offer services commercially. The author can personally recommend *Com Webhosting* of Cardiff who have provided reliable service and good technical support to our students over a number of years (www.comwebhosting.co.uk).
- Web pages and related files are uploaded to a web server by means of an FTP client program. A variety of free software is available, with FileZilla being a popular choice.
- All programming can be carried out with a plain text editor such as Microsoft Notepad. However, it can be an advantage to use a specialist programming text editor such as EditPlus. This utility applies colour coding to different language components such as commands, variables and text strings, and can highlight matching bracket pairs.
- PHP language is interpreted on the server, whilst a JavaScript interpreter is built into the web browser software. Files needed to implement the p5.js high level extension of JavaScript can be freely downloaded from the developers' website (<https://p5js.org>).
- Graphics images are required for a number of the projects. These can be prepared using any simple graphics or desk top publishing software such as Microsoft Word, PowerPoint or Paint, or a specialist graphics application such as Paintshop or Photoshop.
- Design diagrams and flowcharts for project documentation can be prepared with desk top publishing software. However, a quicker and easier alternative which produces very professional results is to use the on-line Lucidchart system (www.lucidchart.com).

In conclusion, the practical programming project can be an exciting and rewarding culmination of a computing course, and can open up new career interests for students. It is very much hoped that independent project work on topics selected by students themselves will continue to form a key component of computer science Foundation Degrees and courses at A-level.